



TESIS - KI2502

Studi Kinerja AODV-PNT dengan Weight Factor Dinamis pada Lingkungan VANET

KUKUH INDRAYANA

NRP. 5114201048

DOSEN PEMBIMBING:

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc

PROGRAM MAGISTER
BIDANG KEAHLIAN KOMPUTASI BERBASIS JARINGAN
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017



TESIS - KI2502

Performance Study of AODV-PNT using Dynamic Weight Factor on VANET Environment

KUKUH INDRAYANA

NRP. 5114201048

SUPERVISOR:

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc

MAGISTER PROGRAMME
NET CENTRIC COMPUTING
INFORMATICS ENGINEERING DEPARTMENT
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2017

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya

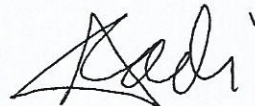
oleh:
KUKUH INDRAYANA
Nrp. 5114201048

Dengan judul :
STUDI KINERJA AODV-PNT DENGAN WEIGHT FACTOR DINAMIS PADA
LINGKUNGAN VANET

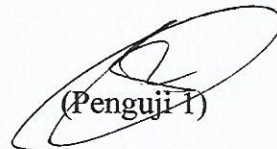
Tanggal Ujian : 17-7-2017
Periode Wisuda : 2016 Genap

Disetujui oleh:

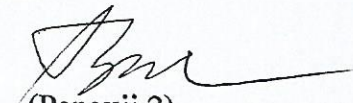
Dr.Eng. Radityo Anggoro, S.Kom, M.Sc
NIP. 1984101620081210002


(Pembimbing 1)

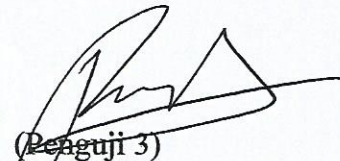
Tohari Ahmad, S.Kom, MIT, Ph.D
NIP. 197505252003121002


(Penguji 1)

Prof.Ir.Supeno Djanali, M.Sc, Ph.D
NIP. 194806191973011001


(Penguji 2)

Royyana Muslim I, S.Kom, M.Kom, Ph.D
NIP. 197708242006041001


(Penguji 3)


Dekan Fakultas Teknologi Informasi,
Dr. Agus Zainal Arifin, S. Kom., M. Kom.
NIP. 197208091995121001

[halaman ini sengaja dikosongkan]

STUDI KINERJA AODV-PNT DENGAN WEIGHT FACTOR DINAMIS PADA LINGKUNGAN VANET

Nama Mahasiswa : Kukuh Indrayana
NRP : 5114201048
Pembimbing : Dr.Eng. Radityo Anggoro S.Kom., M.Sc.

ABSTRAK

Vehicular Ad Hoc Network (VANET) mempunyai karakteristik yang berbeda dari jaringan ad-hoc lainnya. Karakteristik tersebut antara lain adalah tingkat kecepatan serta percepatan sebuah node, variasi jalan dan bangunan yang membatasi mobilitas, dan perbedaan lokasi geografis. Topologi VANET juga dapat berubah secara drastis. VANET membutuhkan protokol routing yang reaktif untuk menangani karakteristik tersebut.

AODV-PNT (AODV-Predicting Node Trend) adalah modifikasi dari protokol routing AODV pada Mobile Ad Hoc Network (MANET) untuk menyesuaikan diri dengan karakteristik VANET. AODV-PNT memodifikasi AODV dengan perhitungan bobot dan prediksi stabilitas node untuk pemilihan rute. Perhitungan bobot dilakukan pada faktor-faktor yang mempengaruhi dalam pemilihan rute. Faktor-faktor tersebut adalah kecepatan node, percepatan, arah, serta kualitas komunikasi antar node.

Pada penelitian ini, penulis mengusulkan empat cara untuk meningkatkan performa AODV-PNT. Pertama adalah penambahan faktor jarak antara next-hop node dengan node tujuan sebagai faktor yang mempengaruhi pemilihan rute. Kedua adalah menetapkan konstanta bobot tiap-tiap faktor melalui pengujian berdasarkan variasi tingkat mobilitas dan kepadatan node. Ketiga adalah menentukan nilai ambang batas untuk perubahan nilai masing-masing faktor dalam prediksi stabilitas node, bukan untuk perubahan nilai total semua faktor seperti pada AODV-PNT. Keempat adalah membedakan pembobotan faktor arah dengan cara menentukan apakah dua buah node bergerak saling mendekat atau menjauh.

Hasil percobaan menunjukkan bahwa metode yang diusulkan menghasilkan packet delivery ratio (PDR), average delay, dan routing overhead (RO) yang lebih baik dibandingkan dengan AODV-PNT. PDR yang dihasilkan lebih tinggi 9.4% untuk skenario grid dan 5.6% untuk skenario riil. Delay yang dihasilkan lebih rendah 0.52 dan 0.08 detik. RO yang dihasilkan lebih rendah 349 dan 133 paket. Metode yang diusulkan menghasilkan PDR, delay, dan RO yang lebih baik karena memilih rute yang lebih stabil berdasarkan modifikasi-modifikasi yang telah dilakukan.

Kata kunci: AODV, pemilihan rute, protokol routing, VANET.

[halaman ini sengaja dikosongkan]

Performance Study of AODV-PNT using Dynamic Weight Factor on VANET environment

Student Name : Kukuh Indrayana
NRP : 5114201048
Supervisor : Dr.Eng. Radityo Anggoro S.Kom., M.Sc.

ABSTRACT

Vehicular Ad Hoc Network (VANET) possess different characteristics from other ad-hoc networks. Those characteristics including speed and acceleration level of node, road and building variation that limit mobility, and geographic location differences. VANET topology also may changes drastically. VANET needs a reactive routing protocol to handle those characteristics.

AODV-PNT(AODV-Predicting Node Trend) is a modification of AODV routing protocol in *Mobile Ad Hoc Network* (MANET) to adapt with characteristics of VANET. AODV-PNT modify AODV by calculating weight and node stability prediction for route selecting. Weight calculation performed on factors that affect in route selection. Those factors are speed of node, acceleration, direction, and communication quality between nodes.

In this research, we propose four ways to improve performances of AODV-PNT. First modification is adding distance between next-hop node and destination node as affecting factor for route selection. Second is setting constant weight for each factors through experiment based on varies node mobility and density. Third is setting threshold value for delta value of each factor in predicting node stability, not total delta value for all factor like in AODV-PNT. Fourth is differ weight of direction factor by determining whether two nodes moving closer or apart each other.

Experiment results shows that the proposed method produces better packet delivery ratio (PDR), average delay, and routing overhead (RO) than AODV-PNT. PDR value produced by the proposed method is higher, 9.4% for grid scenario and 5.6% for real scenario. Delay value is lesser, 0.52 and 0.08 second. RO generated is lesser, 349 and 133 packet. The proposed method generates better PDR, delay and RO because it selects more stable node based on modifications that have been done.

Keywords: AODV, route selection, routing protocol, VANET.

[halaman ini sengaja dikosongkan]

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrohmanirohim.

Alhamdulillahilahirabil'alamin, segala puji bagi Allah Subhanahu Wata'alla, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tesis yang berjudul "*Studi Kinerja AODV-PNT dengan Weight Factor Dinamis pada Lingkungan VANET*" dengan baik dan tepat waktu.

Dalam pelaksanaan dan pembuatan Tesis ini tentunya sangat banyak bantuan-bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Kedua orang tua penulis, yang telah memberikan dukungan moral, spiritual dan material, semangat, perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
3. Bapak Dr. Eng. Radityo Anggoro S.Kom. M.Sc., selaku dosen pembimbing tesis, yang telah memberikan kepercayaan, dukungan, bimbingan, nasehat, perhatian, serta semua yang telah diberikan kepada penulis.
4. Bapak Waskitho Wibisono, S.Kom., M.Eng, Ph.D, Dr.Eng. selaku ketua program studi pasca sarjana jurusan Teknik Informatika ITS, Ibu Dr.Eng. Chastine Fatichah, S.Kom, M.Kom selaku sekretaris prodi S2.
5. Seluruh teman Teknik Informatika ITS angkatan 2008 dan teman pasca sarjana 2014, selama bersama kalian sadar atau tidak telah membentuk karakter dan kepribadian penulis.
6. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu terselesaikannya Tesis ini.

Kesempurnaan tentu sangat jauh tercapai pada Tesis ini, maka penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juli 2017

Kukuh Indrayana

[halaman ini sengaja dikosongkan]

DAFTAR ISI

ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR LAMPIRAN	xxi

BAB 1 PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	4
1.6 Kontribusi Penelitian	4

BAB 2 KAJIAN PUSTAKA

2.1. Vehicular Ad Hoc Network (VANET).....	5
2.1.1 Ad Hoc Routing.....	6
2.1.2 Position Based Routing	6
2.1.3 Cluster Based Routing.....	7
2.1.4 Broadcast Routing	7
2.1.5 Geocast Routing	8
2.2. Ad Hoc On-Demand Distance Vector (AODV).....	8

2.3. AODV – Predicting Node Trend (AODV-PNT)	11
2.3.1 Perubahan metrik routing dan perhitungan TWR	11
2.3.2 Prediksi future TWR	13
2.3.3 Proses Routing	15
2.4. Simulation of Urban Mobility (SUMO)	15
2.5. Network Simulator NS-2	16

BAB 3 METODE PENELITIAN

3.1 Perancangan modifikasi algoritma AODV-PNT.....	17
3.1.1 Faktor yang mempengaruhi bobot pemilihan rute	21
3.1.2 Koefisien bobot tiap faktor	22
3.1.3 Ambang batas untuk penentuan stabilitas node	23
3.1.4 Perbedaan arah node.....	24
3.2 Perancangan skenario simulasi.....	26
3.3 Perancangan metrik analisis	27
3.4 Uji coba dan evaluasi	28

BAB 4 IMPLEMENTASI

4.1 Implementasi skenario grid	31
4.2 Implementasi skenario riil	32
4.3 Implementasi modifikasi AODV-PNT.....	34
4.3.1 Pengaktifan pesan Hello	34
4.3.2 Penambahan atribut pada paket Hello, kelas AODV, dan kelas AODV_Neighbor	34
4.3.3 Modifikasi berkas aodv.cc.....	35
4.3.4 Perhitungan TWR dan future TWR.....	37
4.4 Implementasi metrik analisis.....	39

BAB 5 UJI COBA DAN EVALUASI

5.1 Lingkungan uji coba.....	41
5.2 Uji coba pencarian nilai bobot optimal masing-masing faktor	44
5.3 Uji coba pembedaan faktor arah berdasarkan menjauh atau mendekatnya node.....	46
5.4 Uji coba pencarian nilai ambang batas optimal	48
5.5 Uji coba skenario grid.....	50
5.6 Uji coba skenario riil.....	56

BAB 6 KESIMPULAN DAN SARAN

6.1 Kesimpulan	61
6.2 Saran.....	62

DAFTAR PUSTAKA	60
----------------------	----

LAMPIRAN.....	67
---------------	----

BIODATA PENULIS	73
-----------------------	----

[halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 3.1. Ilustrasi proses route discovery pada AODV	17
Gambar 3.2. Ilustrasi proses route discovery pada AODV-PNT	18
Gambar 3.3. Diagram alir proses route reqeues pada AODV-PNT.....	18
Gambar 3.4. Alur proses penelitian.....	29
Gambar 4.1. Screenshot dari openstreetmap.org	32
Gambar 4.2. Screenshot dari JOSM editor.....	33
Gambar 4.3. Screenshot dari SUMO-GUI	33
Gambar 4.4. Penambahan atribut pada berkas aodv_packet.h.....	34
Gambar 4.5. Modifikasi berkas aodv.h	35
Gambar 4.6. Modifikasi berkas aodv_rtable.h	35
Gambar 4.7. Pseudocode modifikasi fungsi sendHello	36
Gambar 4.8. Modifikasi pada fungsi recvHello	36
Gambar 4.9. Pseudocode perhitungan TWR.....	37
Gambar 4.10. Pseudocode perhitungan future TWR	37
Gambar 5.1. Grafik pencarian nilai bobot optimal faktor jarak pada variasi skenario 2	45
Gambar 5.2. Grafik perbandingan PDR bobot tetap dan dibedakan.....	46
Gambar 5.3. Grafik perbandingan Delay bobot tetap dan dibedakan	47
Gambar 5.4. Grafik perbandingan Overhead bobot tetap dan dibedakan	47
Gambar 5.5. Grafik pencarian nilai ambang batas optimal faktor jarak pada variasi skenario 3.....	49
Gambar 5.6. Grafik perbandingan PDR skenario grid.....	51
Gambar 5.7. Grafik perbandingan Delay skenario grid	52
Gambar 5.8. Grafik perbandingan Overhead skenario grid	52
Gambar 5.9. Contoh trace file	53

Gambar 5.10. Grafik perbandingan jumlah pergantian rute	55
Gambar 5.11. Grafik perbandingan durasi bertahanya rute.....	56
Gambar 5.12. Grafik perbandingan PDR skenario riil	57
Gambar 5.13. Grafik perbandingan Delay skenario riil	57
Gambar 5.14. Grafik perbandingan jumlah pergantian rute	58

DAFTAR TABEL

Tabel 2.1. Penilaian kelayakan relay node.....	14
Tabel 2.2. Hello message pada AODV-PNT	15
Tabel 3.1. Metamorfosa Protokol AODV untuk lingkungan VANET	19
Tabel 3.2. Perbedaan metode yang diusulkan dengan AODV-PNT	20
Tabel 3.3. Variasi tingkat densitas dan mobilitas	23
Tabel 3.4. Tabel stabilitas rute	28
Tabel 4.1. Urutan perintah SUMO untuk membuat skenario grid.....	31
Tabel 5.1. Spesifikasi perangkat keras	41
Tabel 5.2. Spesifikasi perangkat lunak	41
Tabel 5.3. Spesifikasi parameter simulasi NS-2	42
Tabel 5.4. Variasi skenario berdasarkan tingkat mobilitas dan densitas.....	42
Tabel 5.5. Daftar uji coba.....	43
Tabel 5.6. Hasil lengkap nilai bobot optimal tiap faktor.....	45
Tabel 5.7. Hasil lengkap nilai ambang batas optimal tiap faktor	50
Tabel 5.8. Durasi bertahanannya rute pada AODV	54
Tabel 5.9. Durasi bertahanannya rute pada AODV-PNT	54
Tabel 5.10. Durasi bertahanannya rute pada metode yang diusulkan.....	55

[halaman ini sengaja dikosongkan]

/

DAFTAR LAMPIRAN

Lampiran 1. Kode sumber modifikasi fungsi sendHello	67
Lampiran 2. Kode sumber perhitungan TWR.....	67
Lampiran 3. Kode sumber perhitungan future TWR	68
Lampiran 4. Kode sumber perhitungan faktor arah	69
Lampiran 5. Kode sumber penerapan nilai ambang batas	70
Lampiran 6. Skrip AWK packet delivery ratio	70
Lampiran 7. Skrip AWK routing overhead.....	70
Lampiran 8. Skrip AWK average delay	71

[halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komunikasi dan informasi memungkinkan integrasi perangkat telekomunikasi nirkabel dengan kendaraan. Jaringan komunikasi perangkat antar kendaraan disebut dengan *Vehicular Ad Hoc Network* (VANET). VANET merupakan bagian penting dari sistem transportasi cerdas [1]. VANET dapat menyediakan informasi lalu-lintas seperti kecelakaan atau kemacetan [2]. VANET juga dapat memandu kendaraan ketika berada pada persimpangan. Seiring dengan semakin meningkatnya jumlah kendaraan, kebutuhan komunikasi antar kendaraan semakin tinggi. Hal ini mendorong semakin banyaknya riset mengenai VANET.

VANET mempunyai beberapa karakteristik yang berbeda dari jaringan ad-hoc lainnya. Karakteristik tersebut antara lain adalah tingkat kecepatan serta percepatan sebuah *node*, variasi jalan dan bangunan yang membatasi mobilitas, dan perbedaan lokasi geografis[3]. Topologi VANET juga dapat berubah secara drastis, misalnya jika terjadi kemacetan lalu lintas[4]. VANET membutuhkan protokol routing yang reaktif untuk menangani karakteristik tersebut.

Salah satu protokol routing yang dapat digunakan pada VANET adalah Ad-hoc On-demand Distance Vector (AODV)[5]. AODV adalah protokol routing dari Mobile Ad-hoc Networks (MANET). Pada AODV, pencarian rute hanya akan dibuat oleh sebuah *node* apabila *node* tersebut menerima permintaan routing. Struktur topologi dan isi dari tabel routing dibuat setelah menerima permintaan. Mengaplikasikan AODV tanpa modifikasi pada VANET akan menghasilkan performa yang buruk disebabkan oleh perbedaan karakteristik dari MANET dan VANET[6].

Untuk menyesuaikan AODV dengan karakteristik VANET, para periset telah melakukan berbagai modifikasi pada protokol routing AODV. Yu, Guo, dan Wong menambahkan informasi pergerakan dari kendaraan pada proses pencarian rute dan mengusulkan beberapa parameter seperti *total weight* dan *link survival time*[3]. Protokol routing ini dinamakan AODV-VANET. J. Rak mengusulkan CBM-AODV untuk memperbaiki masalah reliabilitas transmisi [7]. Namun, karakteristik utama dari VANET adalah topologi jaringan yang cepat berubah. Kedua protokol diatas tidak mempertimbangkan apakah sebuah *node* stabil atau tidak di waktu yang akan datang.

AODV-PNT (AODV - Predicting Node Trend) [8] memodifikasi AODV dengan perhitungan bobot dan prediksi tren node untuk pemilihan rute. Perhitungan bobot dilakukan pada faktor-faktor yang mempengaruhi dalam pemilihan rute. Faktor-faktor tersebut adalah kecepatan *node*, percepatan, arah, serta kualitas komunikasi antar-*node*. Prediksi tren *node* didasarkan pada nilai dari faktor-faktor tersebut di waktu yang akan datang. Dengan metode ini, AODV-PNT dapat mempertimbangkan kestabilan sebuah *node*.

Pada penelitian ini, penulis mengusulkan empat cara untuk meningkatkan performa AODV-PNT. Modifikasi pertama adalah penambahan faktor jarak antara *next-hop node* dengan *node* tujuan sebagai faktor yang mempengaruhi pemilihan rute. Modifikasi kedua adalah menetapkan konstanta bobot tiap-tiap faktor melalui pengujian berdasarkan variasi tingkat mobilitas dan kepadatan node. Modifikasi ketiga adalah menentukan nilai ambang batas untuk perubahan nilai masing-masing faktor dalam prediksi stabilitas node, bukan untuk perubahan nilai total semua faktor seperti pada AODV-PNT. Modifikasi keempat adalah membedakan pembobotan faktor arah dengan cara menentukan apakah dua buah node bergerak saling mendekat atau menjauh. Logikanya, node yang bergerak saling mendekat akan berada dalam jarak komunikasi dalam rentang waktu yang lebih lama. Tujuan dari modifikasi ini

adalah meningkatkan performa AODV-PNT dalam hal *packet delivery ratio*, *average end-to-end delay*, dan *routing overhead*.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana meningkatkan performa protokol routing AODV-PNT berdasarkan *packet delivery ratio*, *average end-to-end delay*, dan *routing overhead* dengan cara memperhitungkan faktor kecepatan, percepatan, arah, jarak, dan kualitas komunikasi antar-*node*.
2. Bagaimana menentukan bobot untuk faktor kecepatan, percepatan, arah, jarak, dan kualitas komunikasi antar-*node* berdasarkan variasi tingkat kepadatan dan mobilitas, serta apakah *next-hop node* dan *node* destinasi saling menjauh atau mendekat.
3. Bagaimana menentukan nilai ambang batas setiap faktor untuk memprediksi stabilitas node pada waktu yang akan datang.

1.3 Batasan Masalah

Penelitian ini mempunyai batasan-batasan sebagai berikut :

1. Proses pengujian menggunakan network simulator NS-2.
2. Perancangan skenario simulasi dibuat menggunakan SUMO.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Meningkatkan performa protokol routing AODV-PNT berdasarkan *packet delivery ratio*, *average end-to-end delay*, dan *routing overhead* dengan cara memperhitungkan faktor kecepatan, percepatan, arah, jarak, dan kualitas komunikasi antar-*node*.

2. Menentukan bobot untuk faktor kecepatan, percepatan, arah, jarak, dan kualitas komunikasi antar-*node* berdasarkan variasi tingkat kepadatan dan mobilitas, serta apakah *next-hop node* dan *node* destinasi saling menjauh atau mendekat.
3. Menentukan nilai ambang batas setiap faktor untuk memprediksi stabilitas *node* pada waktu yang akan datang.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat meningkatkan performa protokol routing AODV-PNT dalam hal *packet delivery ratio*, *average end-to-end delay*, dan *routing overhead* pada lingkungan VANETs.

1.6 Kontribusi Penelitian

Kontribusi dari penelitian ini adalah sebagai berikut:

1. Menambahkan faktor jarak antara *next-hop node* dengan *node* tujuan sebagai faktor yang mempengaruhi pemilihan rute.
2. Menetapkan konstanta bobot tiap-tiap faktor melalui pengujian berdasarkan variasi tingkat mobilitas dan kepadatan *node*.
3. Menetapkan nilai ambang batas untuk perubahan nilai masing-masing faktor dalam prediksi stabilitas *node*, bukan untuk perubahan nilai total semua faktor seperti pada AODV-PNT.
4. Membedakan pembobotan faktor arah dengan cara menentukan apakah dua buah *node* bergerak saling mendekat atau menjauh.

BAB 2

KAJIAN PUSTAKA

Bab ini berisi pembahasan mengenai konsep dan dasar teori yang akan digunakan dalam penelitian. Konsep dan dasar teori yang akan dibahas adalah VANET, AODV, AODV-PNT, SUMO, dan NS-2.

2.1. Vehicular Ad Hoc Network (VANET)

VANET adalah sebuah bentuk jaringan *ad hoc* nirkabel untuk menyediakan komunikasi antar kendaraan[9]. Tujuan utama dari VANET adalah menyediakan konektivitas untuk pengguna yang sedang berada di jalan dan efisiensi komunikasi antar kendaraan sehingga dapat mewujudkan sistem transportasi cerdas. Sistem transportasi cerdas meliputi berbagai aplikasi seperti monitoring kondisi lalu-lintas secara kooperatif, pengaturan arus lalu-lintas, pencegahan terjadinya kecelakaan, dan lain-lain.

Sejak tahun 2002, dengan pesatnya perkembangan dari teknologi jaringan *ad hoc* nirkabel, VANET telah menarik minat para periset baik dari bidang akademis maupun industri. Jumlah penelitian mengenai VANET meningkat drastis. Sejumlah pabrikan otomotif mulai melakukan investasi jaringan komunikasi antar kendaraan (Inter Vehicle Network / IVC). Audi, BMW, DaimlerChrysler, Fiat, Renault, dan Volkswagen bersama-sama membentuk sebuah organisasi non-profit yang disebut Car2Car Communication Consortium (C2CCC) dengan tujuan untuk meningkatkan keamanan lalu lintas jalan dan efisiensi komunikasi antar kendaraan. VANET, meskipun merupakan sebuah bentuk jaringan *ad hoc* nirkabel, memiliki karakteristik unik disebabkan oleh tingginya mobilitas *node* dan kondisi saluran komunikasi yang tidak reliabel. Karakteristik tersebut mengakibatkan banyak permasalahan riset seperti *routing*, *data dissemination*, *data sharing*, dan isu keamanan.

Banyak protokol routing telah dikembangkan untuk Mobile Ad Hoc Network (MANET), dan sebagian diantaranya dapat langsung diaplikasikan pada VANET. Namun, hasil simulasi menunjukkan performa yang buruk disebabkan oleh karakteristik dari cepatnya pergerakan kendaraan dan dinamisnya pertukaran informasi. Telah banyak protokol *routing* yang dikembangkan untuk VANET. Protokol-protokol tersebut dapat diklasifikasikan menjadi lima kelompok: *ad hoc*, *position-based*, *cluster-based*, *broadcast*, dan *geocast routing*.

2.1.1 Ad Hoc Routing

VANET dan MANET mempunyai prinsip dasar yang sama, yaitu tidak bergantung pada infrastruktur tetap untuk komunikasi, dan mempunyai banyak kemiripan, misalnya *bandwidth* yang kecil dan jarak transmisi radio yang pendek. Banyak protokol *routing* dari MANET yang dapat diaplikasikan pada VANET, seperti AODV dan DSR[10]. AODV dan DSR didesain untuk MANET dan tidak menyimpan informasi rute kecuali saat dibutuhkan.

Sejumlah studi telah dilakukan untuk membandingkan performa protokol *routing ad hoc* pada VANET[11][12][13][14][15]. Hasil simulasi menunjukkan bahwa protokol-protokol tersebut memiliki performa yang buruk dikarenakan tingginya mobilitas *node* pada VANET. Pada[15] AODV diimplementasikan pada jaringan yang terdiri atas enam kendaraan sedan. Hasil penelitian menunjukkan bahwa AODV tidak dapat secara cepat menemukan, mempertahankan, dan memperbarui rute panjang pada VANET.

2.1.2 Position-Based Routing

Pergerakan *node* pada VANET biasanya terbatas mengikuti bentuk dari jalan. Jadi, strategi *routing* yang menggunakan informasi lokasi geografis yang didapatkan dari peta jalan atau sistem navigasi dapat diterapkan. Fakta ini didukung oleh sejumlah studi yang membandingkan performa dari *ad hoc routing* (seperti AODV

dan DSR) dengan strategi routing berbasis posisi dalam skenario lalu lintas perkotaan maupun jalan raya[12][13].

Walaupun *node* dalam jaringan dapat menggunakan informasi posisi untuk pengambilan keputusan dalam *routing*, metode ini tetap mempunyai hambatan. Metode ini biasanya menggunakan *greedy routing* untuk mengirimkan paket ke *node* yang secara geografis paling dekat dengan *node* tujuan. Ketika diimplementasikan pada skenario lalu-lintas perkotaan, *greedy routing* seringkali tidak dapat dilakukan karena komunikasi langsung antar *node* tidak bisa terwujud disebabkan oleh adanya hambatan seperti bangunan atau pohon.

2.1.3 Cluster-Based Routing

Dalam *cluster-based routing*, sebuah infrastruktur jaringan virtual dibentuk dengan cara mengkluster *node*. *Node-node* dalam jaringan diklusterisasi untuk meningkatkan skalabilitas. Setiap *cluster* akan mempunyai *cluster-head* yang bertanggung jawab untuk koordinasi di dalam dan antar *cluster*. *Node* didalam *cluster* berkomunikasi via *direct link*. Komunikasi antar *cluster* dilakukan lewat *cluster-head*. Pengklasteran secara stabil adalah kunci untuk membuat infrastruktur ini.

Banyak protokol *routing cluster-based*[16][17][18] yang telah dikembangkan pada MANET. Namun, VANET mempunyai karakteristik yang berbeda disebabkan oleh tingkah laku pengemudi, batasan dalam mobilitas, serta kecepatan yang tinggi. Akibatnya, teknik routing ini tidak stabil jika diterapkan pada VANET.

2.1.4 Broadcast Routing

Broadcast adalah metode routing yang sering digunakan pada VANET, seperti untuk berbagi kondisi lalu-lintas, cuaca, keadaan darurat, dan lain-lain. *Broadcast* juga digunakan untuk protokol *routing unicast* untuk menemukan rute yang efisien ke *node* tujuan.

Cara termudah untuk mengimplementasikan servis *broadcast* adalah dengan cara *flooding*. Melalui *flooding*, tiap *node* akan melakukan *broadcast* ulang pesan ke semua *node* tetangga. *Flooding* menjamin pesan yang dikirimkan akan mencapai setiap *node* dalam jaringan. Performa *flooding* relatif bagus untuk jaringan dengan

jumlah *node* yang sedikit, dan mudah untuk diimplementasikan. Namun, jika jumlah *node* meningkat, performa dari *flooding* menurun drastis. *Bandwidth* yang dibutuhkan untuk satu transmisi pesan *broadcast* meningkat secara eksponensial.

2.1.5 Geocast Routing

Geocast routing[19] pada dasarnya adalah sebuah *location-based multicast routing*. Tujuan dari *geocast routing* adalah untuk mengirimkan paket dari *node* sumber ke semua *node* lain yang berada di dalam daerah geografis yang spesifik (Zone of Relevance / ZOR).

Aplikasi VANET akan mendapatkan keuntungan dari penggunaan *geocast routing*. Contohnya, sebuah kendaraan yang mendeteksi bahwa kendaraan tersebut mengalami kecelakaan dapat mengirimkan laporan terjadinya kecelakaan ke kendaraan-kendaraan lain yang terdekat. Kendaraan yang berada di luar ZOR tidak perlu diberi laporan untuk menghindari reaksi yang tidak perlu.

2.2. Ad Hoc On-demand Distance Vector (AODV)

Algoritma AODV memungkinkan *multihop routing* yang dinamis antar *node-node* yang bergerak untuk membentuk dan mempertahankan sebuah jaringan *ad hoc*. Menggunakan AODV, *node* dapat memperoleh rute ke destinasi baru secara cepat, dan tidak mengharuskan *node* untuk mempertahankan rute ke destinasi yang sedang tidak aktif dalam komunikasi. AODV memungkinkan *node* untuk merespons kerusakan tautan dan perubahan pada topologi jaringan dalam waktu singkat. Algoritma AODV adalah bebas dari *loop* (*loop-free*) serta cepat mencapai konvergensi ketika topologi jaringan berubah.

Salah satu fitur yang membedakan AODV dengan protokol lain adalah penggunaan *destination sequence number* untuk setiap entri rute. *Destination sequence number* dibuat oleh *node* destinasi untuk diikutsertakan bersama informasi rute yang dikirimkan ke *node* yang melakukan *request*. Penggunaan *destination sequence number* memastikan ketiadaan *loop* dan kemudahan untuk diprogram. Jika

terdapat dua pilihan rute ke satu destinasi, *node* yang melakukan *request* akan memilih rute dengan *destination sequence number* terbesar.

Route Requests (RREQ), Route Replies (RREP), dan Route Errors (RERR) adalah tipe pesan pada AODV. Pesan tersebut diterima via UDP. *Node* yang melakukan *request* akan menggunakan alamat IP-nya sebagai alamat IP pembuat pesan.

Selama *node* pembuat *request* dan *node* destinasi mempunyai rute yang valid untuk menuju satu sama lain, AODV tidak mempunyai peranan. Apabila rute untuk destinasi baru diperlukan, *node* akan mem-*broadcast* sebuah RREQ untuk mencari rute ke destinasi. Sebuah rute dapat ditetapkan apabila RREQ mencapai *node* destinasi atau *intermediate node* yang mempunyai rute “cukup baru” ke destinasi. Rute “cukup baru” adalah entri rute valid ke sebuah destinasi yang *sequence number*-nya minimal sama besar dengan yang terdapat dalam RREQ. Setelah rute ke destinasi ditemukan, RREP akan dikirimkan balik ke sumber RREQ.

Setiap *node* memonitor status tautan dari semua *next-hop* dalam sebuah rute. Ketika kerusakan tautan terdeteksi, sebuah pesan RERR digunakan untuk menotifikasi *node* lain bahwa telah terjadi kerusakan tautan. Pesan RERR menunjukkan destinasi-destinasi yang tidak lagi bisa dicapai melalui tautan yang rusak.

AODV adalah sebuah protokol routing, dan berurusan dengan manajemen tabel rute. Informasi tabel rute harus disimpan sekalipun untuk rute yang berumur sangat pendek. AODV menggunakan field-field berikut pada setiap entri tabel rute:

- Alamat IP destinasi.
- Destination sequence number.
- Status dari *destination sequence number* (valid atau tidak valid).
- Antarmuka jaringan.
- Hop-count (jumlah *hop* yang diperlukan untuk mencapai *node* destinasi).

- Next-hop.

Berikut ini adalah terminologi-terminologi yang berkaitan dengan AODV:

- Active / valid route

Active / valid route adalah sebuah rute menuju destinasi yang mempunyai entri tabel rute dengan status valid. Hanya active / valid *route* yang dapat digunakan untuk meneruskan paket data.

- Broadcast

Broadcast adalah menyebarkan paket ke semua alamat IP dengan batasan 255.255.255.255. Paket *broadcast* bisa jadi tidak disebarkan secara buta, namun *broadcast* berguna untuk memungkinkan penyebaran pesan AODV ke keseluruhan jaringan.

- Destination

Sebuah alamat IP sebagai tujuan pengiriman paket data. Sebuah node akan tahu bahwa node tersebut adalah node destinasi untuk sebuah paket data apabila alamat IP-nya tercantum dalam *field* pada IP *header*.

- Forwarding node

Sebuah *node* yang meneruskan paket data yang ditujukan ke *node* lain, dengan cara mengirimkan ulang paket tersebut ke *next-hop node* yang lebih dekat ke destinasi, sesuai rute yang telah ditentukan.

- Forward route

Rute yang dibuat untuk mengirimkan paket data dari *node* yang memulai proses pencarian rute menuju destinasi yang diinginkan.

- Invalid route

Rute yang tidak dapat dipakai lagi, ditandai dengan status *invalid* pada entri tabel rute.

- Originating node

Node yang memulai pesan pencarian rute AODV untuk diproses dan dikirimkan ke *node-node* lain dalam jaringan *ad hoc*.

- Reverse route

Rute yang dibangun untuk meneruskan sebuah paket RREP dari *node* destinasi atau *intermediate node* yang mengetahui rute ke destinasi, kembali ke *originating node*.

- Sequence number

Sebuah angka yang bertambah secara monoton yang terdapat pada setiap *originating node*. Angka ini digunakan oleh *node* lain untuk menentukan keterbaruan informasi yang berasal dari *originating node*.

2.3. AODV - Predicting Node Trend (AODV - PNT)

AODV - PNT[8] adalah modifikasi dari AODV. Modifikasi utama yang dilakukan adalah :

1. Perubahan metrik routing dengan cara penghitungan Total Weight of the Route (TWR). Perhitungan TWR berdasarkan pada informasi pergerakan kendaraan yaitu kecepatan, percepatan, arah, dan kualitas tautan.
2. Prediksi nilai TWR dari sebuah node pada waktu yang akan datang, dan menghitung stabilitas node berdasarkan sebuah nilai ambang batas *W*.

2.3.1 Perubahan metrik routing dan perhitungan TWR

Ada empat faktor utama yang dijadikan parameter dalam metrik routing. Berdasarkan empat faktor tersebut, TWR dihitung untuk menemukan *next-hop node* yang paling optimal. Detail dari empat faktor tersebut adalah sebagai berikut:

1. Kecepatan dan percepatan kendaraan

Semakin besar perbedaan kecepatan dan percepatan antara dua kendaraan akan menghasilkan TWR yang semakin besar. Alasannya adalah untuk mengantisipasi kerusakan tautan dikarenakan oleh perbedaan kecepatan kendaraan. Kendaraan-kendaraan yang bergerak dalam kecepatan dan percepatan yang relatif sama akan berada di dalam jarak komunikasi radio dalam waktu yang lebih lama.

2. Arah pergerakan kendaraan

Secara logika, kendaraan-kendaraan yang bergerak pada arah yang sama akan berada di dalam jarak komunikasi radio dalam waktu yang lebih lama. Karena itu, vektor arah juga menjadi krusial dalam perhitungan TWR. Faktanya, parameter arah adalah sangat penting dalam menentukan pilihan rute. Nilai perbedaan arah didapat dari perhitungan perbedaan sudut arah berkendara.

3. Kualitas tautan antar kendaraan

Parameter lain yang perlu dipertimbangkan adalah kualitas tautan antara node sumber dengan next-hop. Dalam VANET, kendaraan lain, bangunan, dan objek-objek lain dapat mempengaruhi kualitas tautan antar kendaraan.

Berdasarkan[7] kita dapat mendapatkan indeks stabilitas s_{ij} dari tautan (i, j) berdasarkan informasi dari pergerakan kendaraan.

$$s_{ij} = 1 - \frac{\min(\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}; r)}{r}$$

r adalah jarak komunikasi maksimum antar dua node yang berdekatan.

i_x, i_y adalah koordinat dari node i .

j_x, j_y adalah koordinat dari node j .

Nilai kualitas tautan Q adalah $Q = \frac{1}{s_{ij}}$

TWR dari node sumber ke next-hop node dihitung menggunakan persamaan berikut[8]:

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_d \times |\Theta_n - \Theta_d| + f_q \times Q$$

S_n, A_n , dan Θ_n : Kecepatan, percepatan, dan arah dari next-hop node.

S_d, A_d , dan Θ_d : Kecepatan, percepatan, dan arah dari node tujuan.

f_s : bobot pengali kecepatan.

f_a : bobot pengali percepatan.

f_d : bobot pengali arah.

f_q : bobot pengali kualitas tautan.

Q : kualitas tautan antara node sumber dengan next-hop node.

2.3.2 Prediksi future TWR

Karakteristik paling utama dari VANET adalah topologi yang sering berubah. Sebuah *node* dalam topologi bisa jadi akan pergi dalam waktu singkat. Jadi, future TWR (3 sampai 5 detik berikutnya) akan dihitung untuk memastikan *node* yang stabil yang akan dipilih di masa yang akan datang. Future TWR dihitung berdasarkan faktor-faktor berikut ini:

1. Prediksi kecepatan dan percepatan kendaraan

Penelitian ini memakai asumsi bahwa percepatan dari sebuah *node* adalah konstan selama periode waktu perhitungan *future* TWR. Kecepatan *node* berubah sesuai dengan rumus percepatan.

2. Prediksi arah pergerakan kendaraan

Di dunia nyata, jika terdapat tikungan di jalan, kendaraan akan mempunyai akselerasi negatif dan menyalakan lampu sein. Kendaraan yang memenuhi kondisi tersebut akan menghitung vektor arah dengan bantuan GPS.

3. Prediksi kualitas tautan antar kendaraan

Berdasarkan data kecepatan, percepatan, dan arah kendaraan, metode ini akan mendapatkan koordinat dari sebuah *node* di waktu yang akan datang. Kualitas tautan antar *node* dapat dihitung berdasarkan koordinat dari kedua *node* tersebut.

Setelah mendapatkan data tersebut, future TWR dari *node* akan dapat dihitung. Nilai ini akan digunakan sebagai sebuah parameter untuk menentukan apakah sebuah *node* layak untuk dijadikan *relay node*. Selanjutnya, penilain kelayakan sebuah *node* untuk dijadikan *relay node* dijelaskan pada Tabel 2.1.

Tabel 2.1.
Penilaian kelayakan relay node [8]

TWR saat ini	Status	Future TWR	Penilaian
Optimal	Tidak stabil	Lebih baik	Pilih node
Optimal	Stabil		Pilih node
Suboptimal	Tidak stabil	Lebih baik	Pilih node
Subotimal	Stabil		Pilih node
Kondisi lain			Abaikan node

TWR saat ini : semakin kecil nilai TWR adalah semakin bagus. Node dengan nilai TWR paling kecil adalah “Optimal”, sedangkan *node* lain dengan nilai TWR bagus disebut “Suboptimal”.

Status : Sebuah nilai *W* ditentukan sebagai nilai ambang batas. Jika nilai absolut dari perbedaan TWR saat ini dan future TWR (Δ TWR) kurang

dari W , maka node tersebut dianggap sebagai node yang stabil. Jika ΔTWR lebih dari W , maka node tersebut dianggap tidak stabil.

2.3.3 Proses Routing

Dalam AODV, *node* akan menyebarkan *hello message* secara periodik untuk pemeliharaan jalur *routing*. Pada AODV-PNT, *hello message* dimodifikasi agar dapat mengumpulkan informasi pergerakan dari *node*. Dari informasi ini, *node* sumber dapat menghitung TWR, status, dan *future TWR*, lalu memilih himpunan *relay node*. Setelah itu, *node* sumber melakukan *multicast* pesan RREQ ke semua *relay node*. Jika salah satu *relay node* tahu jalur menuju destinasi, *node* tersebut akan mengirim pesan RREP kepada *node* sumber beserta jalur kembalinya. Jika *relay node* tidak tahu jalur ke destinasi, *node* tersebut akan memilih himpunan *relay node* selanjutnya dan melakukan *multicast* RREQ. Proses ini akan diulangi sampai sebuah *node* menemukan jalur menuju destinasi atau destinasi tidak dapat dicapai. *Hello message* pada AODV-PNT ditunjukkan pada Tabel 2.2.

Tabel 2.2.

Hello message pada AODV-PNT [8]

Type	Source ID	Speed	Acceleration	Destination
------	-----------	-------	--------------	-------------

2.4. Simulation of Urban Mobility (SUMO)

SUMO adalah aplikasi sumber terbuka untuk simulasi lalu-lintas. Simulasi yang dilakukan memungkinkan untuk memodelkan berbagai macam topik manajemen lalu-lintas. Simulasi pada SUMO bersifat sangat mendetail: setiap kendaraan dimodelkan secara eksplisit, mempunyai rutenya sendiri dan bergerak secara sendiri-sendiri di dalam jaringan. SUMO menyediakan berbagai pilihan untuk memasukkan unsur variabel acak (*randomness*).

Pengembangan SUMO dimulai pada tahun 2000. Alasan utama dibalik pengembangan SUMO adalah untuk mendukung komunitas riset lalu-lintas dengan sebuah alat yang memiliki kemampuan untuk mengimplementasikan dan

mengevaluasi algoritma. Publikasi terbaru mengenai pengembangan dan aplikasi SUMO ditulis oleh Krajzewicz dkk.[20] pada tahun 2012.

2.5. Network Simulator NS-2

NS-2 adalah sebuah simulator sumber terbuka yang didesain secara spesifik untuk riset dalam jaringan komunikasi komputer[21]. Sejak kemunculannya pada tahun 1989, NS-2 telah mendapatkan perhatian yang besar dari industri, akademisi, dan pemerintah. Setelah melalui penelitian dan perbaikan selama bertahun-tahun, NS-2 sekarang berisi modul-modul untuk berbagai komponen jaringan seperti routing, protokol transport layer, aplikasi, dan lain-lain.

Untuk meneliti performa jaringan, peneliti dapat dengan mudah menggunakan bahasa script untuk mengkonfigurasi sebuah jaringan, dan mengamati hasil yang dikeluarkan oleh NS-2. NS-2 telah menjadi simulator jaringan sumber terbuka yang paling banyak digunakan, dan salah satu simulator jaringan yang paling banyak digunakan.

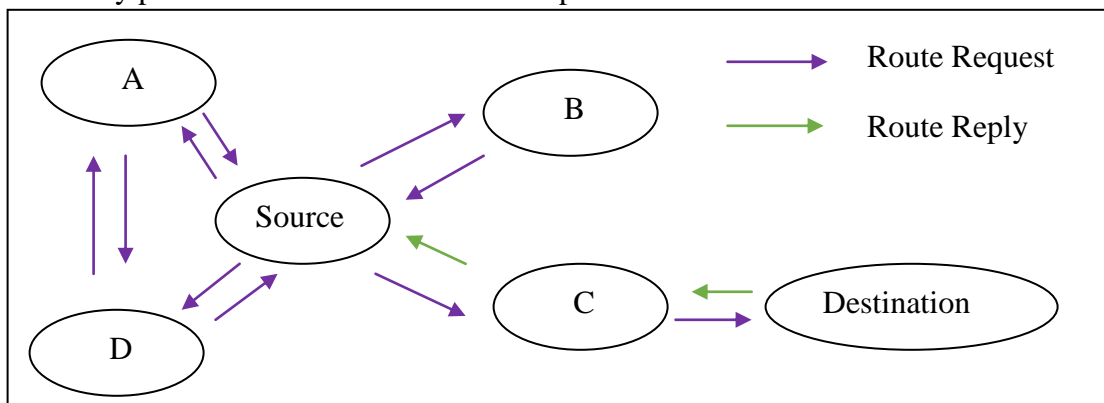
BAB 3

METODE PENELITIAN

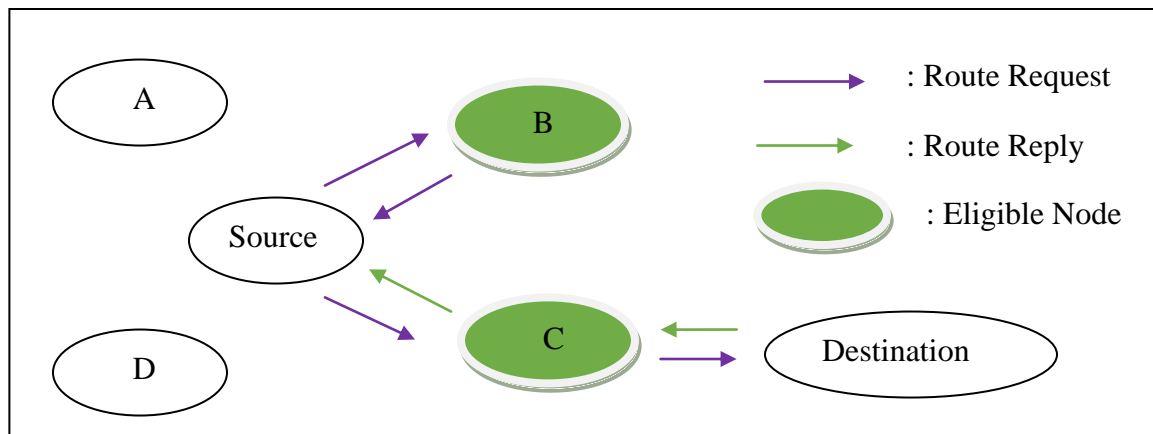
Bab ini berisi uraian mengenai langkah serta jadwal kegiatan penelitian. Langkah penelitian terdiri dari perancangan modifikasi algoritma AODV-PNT, perancangan skenario simulasi, perancangan metrik analisis, uji coba, dan evaluasi.

3.1 Perancangan modifikasi algoritma AODV-PNT

Algoritma yang akan dirancang adalah modifikasi dari AODV-PNT. AODV-PNT sendiri adalah pengembangan dari AODV. Metamorfosa dari protokol AODV hingga AODV-PNT penulis tampilkan pada Tabel 3.1. AODV-PNT berbeda dari AODV pada proses pengiriman route request dan pemilihan next-hop node. Pada AODV, paket request akan di-broadcast ke semua next-hop node. Semua next-hop node kemudian akan melakukan broadcast ulang paket tersebut ke node tetangganya, jika node tersebut belum mengetahui jalur ke node tujuan. Proses route discovery pada AODV diilustrasikan pada Gambar 3.1. Berbeda dengan AODV, AODV-PNT menyeleksi next-hop node yang akan dikirim paket request. AODV-PNT mengumpulkan informasi pergerakan node tetangga dari paket hello. Node tetangga yang sudah diseleksi akan dimasukkan ke daftar eligible node. Paket request kemudian akan di-multicast ke node yang ada di dalam daftar saja. Proses route discovery pada AODV-PNT diilustrasikan pada Gambar 3.2.

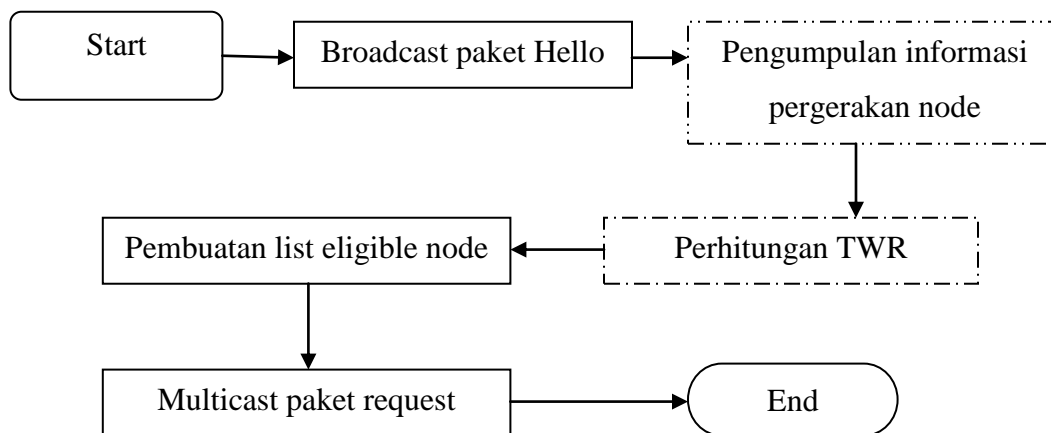


Gambar 3.1. Ilustrasi proses route discovery pada AODV



Gambar 3.2. Ilustrasi proses route discovery pada AODV-PNT

Berdasarkan hasil pengujian, AODV-PNT memiliki performa yang lebih baik daripada AODV, berdasarkan metrik packet delivery ratio, average end-to-end delay, dan routing overhead [8]. Meskipun demikian, AODV-PNT masih memiliki celah untuk dikembangkan lebih lanjut. Pada penelitian ini, penulis melakukan empat modifikasi pada AODV-PNT. Modifikasi tersebut akan diuraikan satu-persatu pada sub-bab berikutnya. Ringkasan dari keseluruhan modifikasi dijabarkan pada Tabel 3.2. Modifikasi yang dilakukan terletak pada proses route request, yaitu pada proses pengumpulan informasi pergerakan node dan perhitungan TWR. Diagram alir proses route request ditunjukkan pada Gambar 3.3.



Gambar 3.3. Diagram alir proses route request pada AODV-PNT

Tabel 3.1.

Metamorfosa Protokol AODV untuk Lingkungan VANETs

Algoritma	Kelebihan	Kekurangan
AODV (2003)	<ul style="list-style-type: none"> - Hanya membentuk rute ke destinasi jika dibutuhkan - Bebas dari <i>infinite loop</i> - Mempunyai nilai <i>routing overhead</i> yang relatif kecil 	<ul style="list-style-type: none"> - Menghasilkan performa yang tidak bagus jika diimplementasikan tanpa modifikasi pada VANET
AODV - VANET (2011)	<ul style="list-style-type: none"> - Memperkenalkan konsep Total Weight of the Route (TWR) untuk pemilihan rute berdasarkan faktor kecepatan, percepatan, arah, dan kualitas tautan - Memprediksi Link Survival Time (LST) dari rute yang dipilih 	<ul style="list-style-type: none"> - Tidak mempertimbangkan kestabilan node pada waktu yang akan datang
CBM - AODV (2013)	<ul style="list-style-type: none"> - Pemilihan rute menggunakan total cost berdasarkan faktor kecepatan dan arah - Menerapkan probabilitas untuk memprediksi reliabilitas transmisi antara dua node 	<ul style="list-style-type: none"> - Mengasumsikan kecepatan node konstan - Tidak mempertimbangkan kestabilan node pada waktu yang akan datang
AODV - PNT (2014)	<ul style="list-style-type: none"> - Mempertimbangkan kestabilan node pada waktu yang akan datang dengan cara menghitung future TWR dan membandingkan ΔTWR dengan nilai ambang batas W 	<ul style="list-style-type: none"> - Tidak mempertimbangkan faktor jarak antara next-hop node dengan destinasi - Koefisien bobot masing-masing faktor konstan untuk berbagai variasi densitas dan mobilitas - Perbedaan arah node

		<p>berdasarkan derajat tidak dapat membedakan apakah dua node menjauh atau mendekat</p> <ul style="list-style-type: none"> - Tidak mempertimbangkan ketidakstabilan node jika salah satu faktor berubah drastis namun total ΔTWR masih dibawah nilai W
--	--	---

Tabel 3.2.

Perbedaan Metode yang diusulkan dengan AODV-PNT

	AODV - PNT	Modifikasi yang diusulkan
Faktor yang mempengaruhi bobot pemilihan rute (TWR).	Perbedaan kualitas tautan (antara next-hop node dengan sumber), kecepatan, percepatan, dan arah (antara next-hop node dan destinasi).	Perbedaan kualitas tautan (antara next-hop node dengan sumber), kecepatan, percepatan, arah, dan jarak (antara next-hop node dan destinasi).
Koefisien bobot masing-masing faktor.	Konstan.	Berubah-ubah sesuai variasi tingkat densitas dan mobilitas.
Ambang batas untuk penentuan stabilitas next-hop node.	Ambang batas diterapkan pada perubahan total bobot (ΔTWR)	Setiap faktor memiliki nilai ambang batas sendiri.
Perbedaan arah node	Perbedaan arah dihitung berdasarkan selisih derajat.	Selain selisih derajat, dua node dinyatakan saling mendekat atau menjauh dilihat dari posisi titik potongnya, dan diberikan koefisien bobot yang berbeda

3.1.1 Faktor yang Mempengaruhi Bobot Pemilihan Rute

AODV-PNT menggunakan empat faktor untuk pembobotan pemilihan rute. Empat faktor tersebut adalah kecepatan, percepatan, arah, dan kualitas tautan. Untuk faktor kualitas tautan, AODV-PNT membandingkan antara *next-hop node* dengan *node* sumber. Untuk faktor lain, AODV-PNT membandingkan antara *next-hop node* dengan *node* tujuan.

Pada penelitian ini, penulis mengusulkan penambahan faktor jarak dalam pemilihan rute. Jarak yang dibandingkan adalah jarak antara *next-hop node* dengan *node* tujuan. Faktor jarak patut ditambahkan karena alasan pembobotan pemilihan rute adalah memilih *node* yang paling “mendekati” *node* tujuan, baik secara kecepatan, percepatan, maupun arah. Jika ada dua *node* dengan bobot TWR yang sama (sebelum perhitungan faktor jarak) tentu akan dipilih *node* dengan jarak yang lebih dekat ke *node* tujuan.

Jarak antar node dihitung menggunakan rumus Euclidean distance, yaitu jarak dua node berdasarkan garis lurus. Pada VANET, memang node bergerak mengikuti ruas jalan. Namun, transmisi komunikasi antar node tidak terpengaruh ruas jalan.

Rumus perhitungan bobot TWR setelah penambahan faktor jarak adalah sebagai berikut :

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_d \times |\Theta_n - \Theta_d| + f_q \times Q \\ + f_{dist} \times Dist$$

S_n , A_n , dan Θ_n : Kecepatan, percepatan, dan arah dari next-hop node.

S_d , A_d , dan Θ_d : Kecepatan, percepatan, dan arah dari node tujuan.

f_s : bobot pengali kecepatan.

f_a : bobot pengali percepatan.

f_d : bobot pengali arah.

f_q : bobot pengali kualitas tautan.

Q : kualitas tautan antara node sumber dengan next-hop node.

f_{dist} : bobot pengali jarak.

Dist : jarak antara next-hop node dengan node tujuan. Jarak dihitung berdasarkan posisi koordinat dari kedua node

3.1.2 Koefisien Bobot tiap Faktor

Pada AODV-PNT koefisien bobot tiap faktor adalah konstan. Nilai koefisien tersebut didapat dari hasil percobaan. Pada penelitian ini penulis berasumsi bahwa koefisien bobot bisa jadi berubah-ubah tergantung pada tingkat kecepatan (mobilitas) dan tingkat kepadatan node (densitas).

Pada penelitian ini, tingkat mobilitas dibagi menjadi tiga kategori yaitu rendah, sedang, dan tinggi. Pembagian kategori didasarkan pada Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 111 Tahun 2015 Tentang Tata Cara Penetapan Batas Kecepatan [22]. Peraturan tersebut menetapkan batas kecepatan untuk jalan kawasan perkotaan, jalan antarkota, dan jalan bebas hambatan. Kategori kecepatan rendah mengacu pada batas kecepatan jalan perkotaan, yaitu maksimal 50 km/jam (± 14 m/s). Kategori kecepatan sedang mengacu pada batas kecepatan jalan antarkota, yaitu maksimal 80 km/jam (± 22 m/s). Kategori kecepatan tinggi mengacu pada batas kecepatan jalan bebas hambatan yaitu maksimal 100 km/jam (± 28 m/s).

Tingkat densitas juga dibagi dalam tiga kategori. Tingkat densitas rendah memiliki *node* sebanyak 30, tingkat densitas sedang 50, dan tingkat densitas tinggi 100. Variasi tingkat densitas dan mobilitas dapat dilihat pada Tabel 3.3.

Tabel 3.3.
Variasi tingkat densitas dan mobilitas

	Rendah	Sedang	Tinggi
Densitas (node)	30	50	100
Mobilitas (m/s) [22]	2 - 14	15 - 22	23 - 28

3.1.3 Ambang Batas Untuk Penentuan Stabilitas Node

Pada AODV-PNT mekanisme penentuan stabil tidaknya sebuah node didasarkan pada perbandingan antara nilai total ΔTWR (selisih mutlak nilai TWR dengan future TWR) dengan nilai ambang batas W . Jika nilai ΔTWR kurang dari W , maka *node* tersebut dinyatakan stabil. Metode seperti ini mempunyai kelemahan, yaitu apabila salah satu faktor mengalami perubahan nilai secara drastis, sedangkan faktor lain cenderung konstan, sebuah node akan tetap dinyatakan stabil. Sebagai contoh, sebuah node dengan nilai Δ kecepatan, percepatan, arah, dan kualitas tautan sebesar 25, 25, 25, 25 adalah stabil jika ambang batas $W > 100$. Demikian pula dengan nilai Δ sebesar 70, 30, 0, 0. Padahal, untuk contoh nilai Δ yang kedua, salah satu faktor berubah secara drastis. Hal seperti ini tentu akan mengakibatkan nilai future TWR selanjutnya dari node tersebut akan menjadi tidak stabil.

Untuk mengatasi permasalahan tersebut, penulis mengusulkan penggunaan nilai ambang batas untuk Δ masing-masing faktor, bukan total ΔTWR . Sebuah node dianggap stabil jika dan hanya jika semua nilai Δ tiap faktor tidak melebihi nilai ambang batas masing-masing.

Faktor yang mempunyai nilai ambang batas hanyalah faktor yang mempunyai nilai future. Faktor-faktor tersebut adalah kecepatan, jarak, dan kualitas tautan. Faktor percepatan dan arah pergerakan tidak mempunyai nilai future karena diasumsikan nilai kedua faktor tersebut tidak berubah dalam rentang waktu perhitungan future TWR.

3.1.4 Perbedaan Arah Node

Pada AODV-PNT perbedaan arah node dihitung berdasarkan selisih derajat arah pergerakan node. Derajat pergerakan node didapat dari fungsi pada SUMO. Metode seperti ini tidak dapat membedakan apakah dua buah node bergerak saling mendekat atau menjauh berdasarkan arahnya. Dua node yang berbeda arah bisa jadi saling menjauh, bisa pula mendekat.

Pada penelitian ini penulis mengusulkan metode pencarian titik potong dua garis untuk menentukan apakah dua node saling mendekat atau menjauh. Logikanya, node yang bergerak saling mendekat akan berada dalam jarak komunikasi dalam rentang waktu yang lebih lama. Untuk dua node yang saling menjauh akan diberikan faktor pengali bobot dua kali lipat dari dua node yang saling mendekat. Formula perhitungan TWR akan ditambahkan menjadi:

$$f_d = \text{saling_mendekat}(\text{node1}, \text{node2}) ? f_d : f_d * 2$$

f_d : bobot pengali arah.

`saling_mendekat()` : fungsi untuk menentukan apakah dua node saling mendekat.

Langkah-langkah untuk menentukan apakah dua node saling mendekat atau menjauh adalah sebagai berikut.

1) Pencarian sudut terhadap garis vertikal

Sudut yang diperoleh dari fungsi SUMO adalah sudut berdasarkan garis horisontal dan dalam satuan radian. Langkah pertama adalah mengubah satuan radian ke derajat. Konversi dari radian ke derajat dilakukan menggunakan persamaan berikut.

$$1 \text{ radian} = \frac{180^\circ}{\pi}$$

Langkah berikutnya adalah mencari sudut berdasarkan garis vertikal. Jika sudut horisontal diantara -90° sampai 90° , maka sudut vertikal adalah

90 – sudut horizontal. Jika tidak, maka sudut vertical adalah 270 – sudut horizontal.

2) *Pencarian gradien / kemiringan*

Satu node dengan arahnya akan dianggap bergerak melalui sebuah garis lurus. Gradien dari garis tersebut dapat dicari berdasarkan sudut terhadap garis horizontal. Jika sudut terhadap garis horizontal adalah Θ dan gradien adalah m , persamaan untuk gradien adalah sebagai berikut.

$$m = \tan \Theta$$

3) *Pencarian persamaan garis lurus*

Apabila gradien serta posisi sebuah node sudah diketahui, maka persamaan garis lurus yang melewati node tersebut dapat diketahui. Untuk x_1 dan y_1 sebagai koordinat node, dan m sebagai gradien, rumus persamaan garis adalah sebagai berikut.

$$y - y_1 = m (x - x_1)$$

4) *Pencarian letak titik potong*

Dua buah node dianggap bergerak melalui sebuah garis lurus dengan lintasan masing-masing. Untuk dapat ditentukan apakah pergerakan keduanya saling mendekat atau menjauh perlu dicari letak titik potongnya. Jika terdapat dua persamaan garis sebagai berikut,

$$y = m_1x + c_1$$

$$y = m_2x + c_2$$

maka koordinat sumbu x dari titik potongnya adalah

$$x = (c_2 - c_1) / (m_1 - m_2)$$

sedangkan koordinat sumbu y dapat dicari dengan memasukkan nilai x ke salah satu persamaan garis.

5) *Saling mendekat atau saling menjauh*

Dua buah node dinyatakan saling mendekat jika dan hanya jika keduanya bergerak mendekati titik potong. Untuk Θ sebagai sudut terhadap garis vertical, definisi mendekati titik potong adalah berdasarkan ketentuan berikut.

- $\Theta < 90^\circ$: Nilai x dan y dari titik potong harus lebih besar daripada nilai y dari node.
- $90^\circ \leq \Theta < 180^\circ$: Nilai x dari titik potong harus lebih besar daripada nilai x dari node. Nilai y dari titik potong harus lebih kecil atau daripada nilai y dari node.
- $180^\circ \leq \Theta < 270^\circ$: Nilai x dan y dari titik potong harus lebih kecil daripada nilai y dari node.
- $270^\circ \leq \Theta < 360^\circ$: Nilai x dari titik potong harus lebih kecil daripada nilai x dari node. Nilai y dari titik potong harus lebih besar atau daripada nilai y dari node.

3.2 Perancangan Skenario Simulasi

Algoritma yang telah dirancang akan diimplementasikan pada simulator NS-2. Ada dua skenario yang akan digunakan, yaitu skenario riil dan skenario grid. Skenario riil diambil dari peta lalu lintas kota Surabaya menggunakan aplikasi OpenStreetMap. OpenStreetMap adalah proyek kolaborasi pembuatan peta dunia yang bebas disunting. Peta dari OpenStreetMap akan disunting menggunakan Java Open Street Map Editor (JOSM). Skenario grid dibuat menggunakan aplikasi SUMO.

Setelah berkas peta didapat, baik untuk skenario grid maupun riil, implementasi algoritma yang telah dirancang akan disimulasikan pada SUMO. Hasil simulasi dianalisis menggunakan skrip AWK untuk menghitung metrik analisis yang terdiri atas *packet delivery ratio*, *average end-to-end delay*, *routing overhead*, dan *route stability*. Metrik analisis akan dibandingkan antara algoritma yang telah dirancang dengan algoritma AODV-PNT.

3.3 Perancangan Metrik Analisis

Penelitian ini menggunakan empat metrik analisis untuk mengukur performa algoritma yang telah dirancang. Metrik-metrik tersebut adalah *packet delivery ratio*, *average end-to-end delay*, *routing overhead*, dan *route stability*.

Packet delivery ratio adalah perbandingan dari jumlah paket yang dikirimkan dan diterima. Rumus *packet delivery ratio* (PDR) adalah sebagai berikut.

$$PDR = \frac{Data_{diterima}}{Data_{terkirim}}$$

Average end-to-end delay merupakan rata-rata waktu yang dibutuhkan oleh semua paket untuk sampai ke tujuan. Paket yang diperhitungkan hanya paket yang mencapai tujuan. Rumus *average end-to-end delay* adalah sebagai berikut.

$$Delay = \frac{\sum_{i=0}^n t_{received}[i] - t_{sent}[i]}{pktCounter}$$

i adalah indeks paket yang sampai ke tujuan.

$t_{received}$ adalah waktu sampainya paket.

t_{sent} adalah waktu dikirimkannya paket.

$pktCounter$ adalah jumlah total paket yang sampai ke tujuan.

Routing overhead adalah jumlah paket routing yang ditransmisikan. Paket routing yang ditransmisikan terdiri dari RREQ, RREP, dan RERR. Rumus dari *Routing overhead* (RO) adalah sebagai berikut.

$$RO = RREQ_{sent} + RREP_{sent} + RERR_{sent}$$

Route stability dihitung berdasarkan berapa kali rute berubah selama proses pengiriman data dari node sumber ke node tujuan. Pengukuran stabilitas rute direpresentasikan dalam bentuk tabel yang terdiri atas jumlah rute dan durasi bertahannya rute. Contoh dari tabel tersebut dapat dilihat pada Tabel 3.4.

Tabel 3.4.

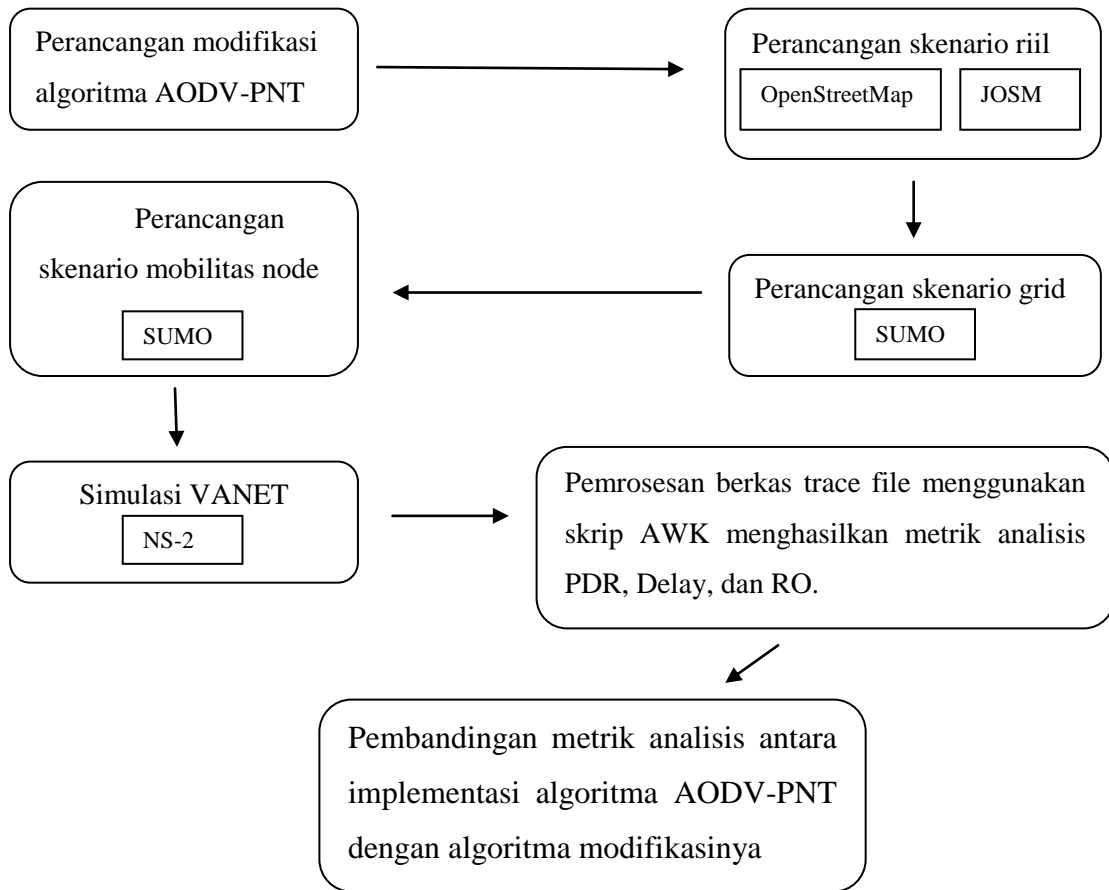
Tabel Stabilitas Rute

Nomor Rute	Durasi (detik)
1	0.099
2	1.151
3	0.876
4	3.532
5	4.442
6	2.234
7	7.721

3.4 Uji Coba dan Evaluasi

Setelah tahapan perancangan algoritma selesai, tahapan selanjutnya adalah ujicoba. Algoritma yang telah dirancang akan diimplementasikan pada simulator NS-2. Masukan untuk simulasi adalah skenario uji coba yang telah dibuat menggunakan SUMO. Selain algoritma yang telah dirancang, algoritma AODV-PNT juga akan diimplementasikan untuk melakukan perbandingan performa.

Keluaran dari proses simulasi adalah berkas *trace file* dengan ekstensi *.tr*. Berkas trace file akan diproses menggunakan skrip AWK untuk menghitung metrik analisis. Alur proses penelitian mulai dari perancangan sampai dengan uji coba ditunjukkan pada Gambar 3.4.



Gambar 3.4. Alur proses penelitian

[halaman ini sengaja dikosongkan]

BAB 4

IMPLEMENTASI

4.1. Implementasi Skenario Grid

Ukuran dari skenario grid adalah 800 meter x 800 meter dengan panjang jalan masing-masing 200 meter. Kecepatan kendaraan yang diperbolehkan diatur sesuai dengan variasi tingkat mobilitas, yaitu 2-14 meter/detik untuk variasi mobilitas rendah, 15-22 meter/detik untuk variasi mobilitas sedang, dan 23-28 meter/detik untuk mobilitas tinggi. Perintah SUMO untuk membuat skenario grid adalah `netgenerate`. Keluaran dari perintah ini adalah berkas peta `map.net.xml`. Perintah-perintah SUMO untuk membuat skenario simulasi grid secara berurutan dapat dilihat pada Tabel 4.1.

Tabel 4.1. Urutan perintah SUMO untuk membuat skenario grid

Perintah	Masukan	Keluaran
<code>netgenerate</code>	-	<code>map.net.xml</code>
<code>randomTrips.py</code>	<code>map.net.xml</code>	<code>trips.trips.xml</code>
<code>duarouter</code>	<code>trips.trips.xml</code>	<code>route.rou.xml</code>
<code>sumo</code>	<code>map.net.xml</code> dan <code>route.rou.xml</code>	<code>scenario.xml</code>
<code>traceExporter.py</code>	<code>scenario.xml</code>	<code>scenario.tcl</code>

Setelah peta `map.net.xml` terbentuk, dilakukan pembuatan titik asal dan tujuan kendaraan menggunakan perintah `randomTrips.py`. Salah satu parameter `randomTrips.py` adalah jumlah node. Jumlah node diatur sesuai dengan variasi tingkat densitas, yaitu 30 untuk densitas rendah, 50 untuk densitas sedang, dan 100 untuk densitas tinggi. Keluaran dari perintah `randomTrips.py` adalah berkas `trips.trips.xml`.

Setelah titik asal dan tujuan didefinisikan, dilakukan pembuatan rute yang akan digunakan oleh kendaraan menggunakan perintah `duarouter`. Selanjutnya dilakukan pembuatan berkas `file.sumocfg` yang akan digunakan sebagai argument

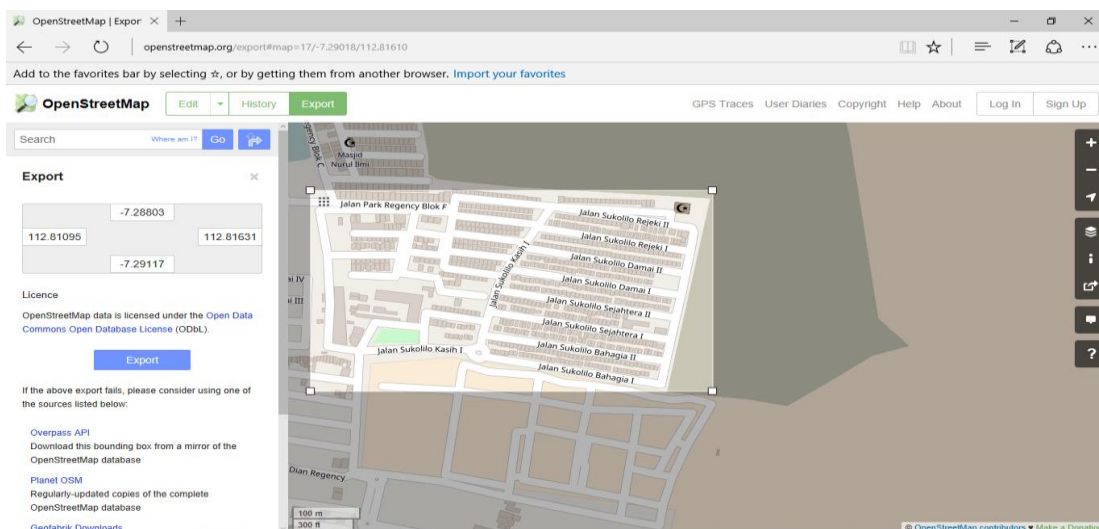
perintah sumo. Berkas file.sumocfg diperlukan untuk mendefinisikan letak berkas map.net.xml dan route.rou.xml.

Keluaran dari perintah sumo adalah berkas scenario.xml. Berkas ini lalu dikonversi ke format NS-2 menggunakan perintah traceExporter.py. Hasil konversi adalah berkas scenario.tcl yang dapat dijalankan pada NS-2.

4.2. Implementasi Skenario Riil

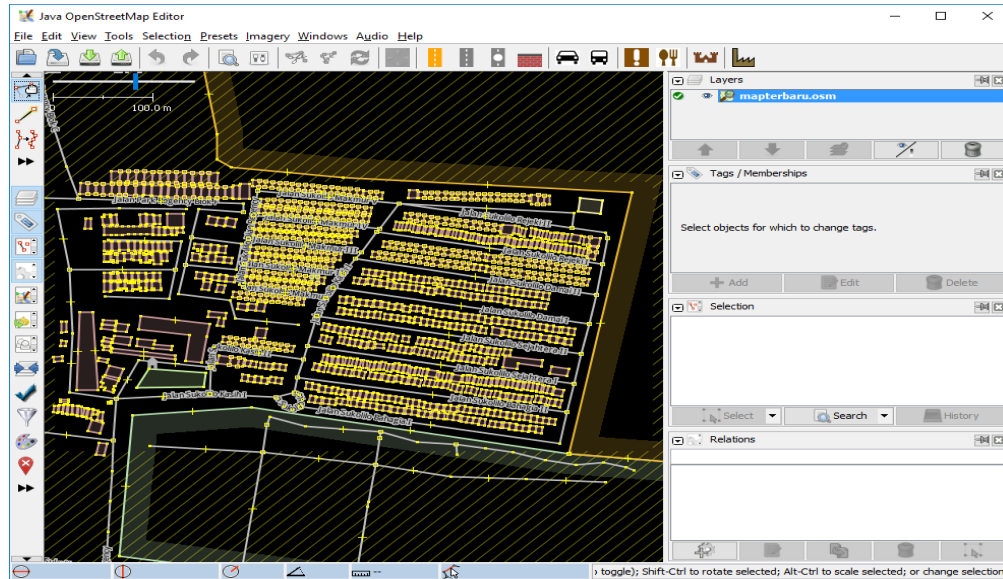
Skenario riil memakai peta yang diambil dari OpenStreetMap. Caranya adalah mengakses situs openstreetmap.org, memasukkan nama daerah yang akan diambil petanya, klik tombol export, klik pilihan “manually select a different area” lalu drag mouse untuk mengambil daerah yang kita inginkan. Screenshot dari OpenStreetMap ditunjukkan pada Gambar 4.1.

Hasil dari proses export pada OpenStreetMap adalah berkas dengan ekstensi .osm. Berkas ini dapat disunting menggunakan Java Open Street Map (JOSM) Editor. Tujuan penyuntingan adalah menghapus atau menambahkan jalan agar tidak ada ruas pada peta yang jarang dikunjungi kendaraan. Screenshot dari JOSM Editor dapat dilihat pada Gambar 4.2.

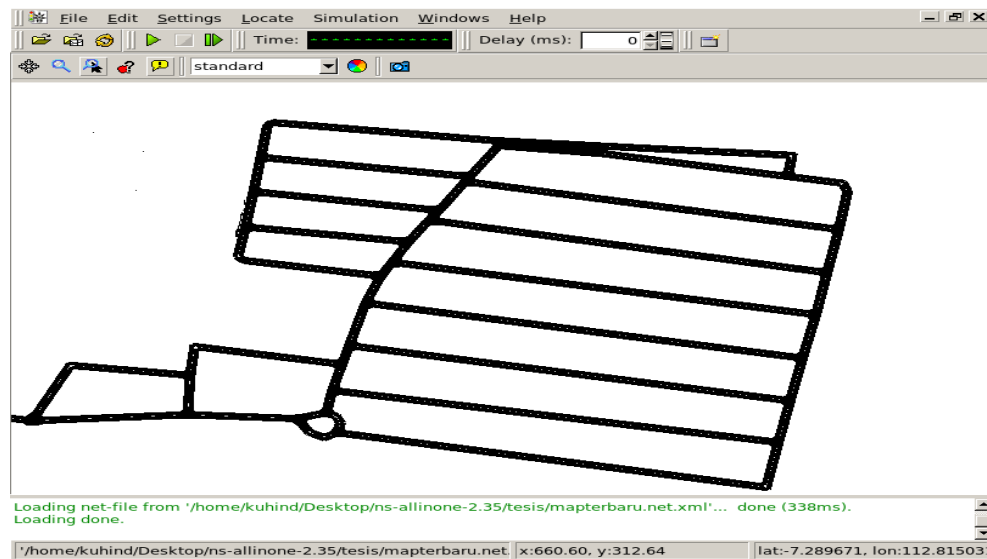


Gambar 4.1. Screenshot dari openstreetmap.org

Peta hasil penyuntingan lalu diubah menjadi berkas .net.xml agar dapat diproses oleh SUMO. Konversi dari .osm menjadi .net.xml menggunakan perintah netconvert pada SUMO. Berkas .net.xml selanjutnya dapat diproses dengan cara yang sama seperti pada skenario grid. Berkas .net.xml dapat dibuka menggunakan SUMO-GUI seperti ditampilkan pada Gambar 4.3.



Gambar 4.2. Screenshot dari JOSM Editor



Gambar 4.3. Screenshot dari SUMO-GUI

4.3. Implementasi Modifikasi AODV-PNT

Protokol AODV-PNT memodifikasi protokol AODV pada beberapa bagian. Modifikasi yang dilakukan antara lain terletak pada pengaktifkan pesan Hello, penambahan atribut pada paket Hello, kelas AODV, dan kelas AODV_Neighbor, modifikasi pada fungsi sendHello dan recvHello dalam berkas aodv.cc, serta penghitungan nilai TWR dan future TWR.

4.3.1 Pengaktifan pesan Hello

Pesan Hello pada protokol AODV di NS-2 secara default tidak diaktifkan. Pesan Hello dapat diaktifkan dengan cara memberi *comment* pada baris AODV_LINK_LAYER_DETECTION (baris ke-58) dan AODV_USE_LL_METRIC (baris ke-66) berkas aodv.h.

4.3.2 Penambahan atribut pada paket Hello, kelas AODV, dan kelas AODV_Neighbor

Pada AODV-PNT, informasi nilai faktor kecepatan, percepatan, lokasi, dan arah dari sebuah node harus ditambahkan pada paket Hello. Informasi-informasi tersebut disimpan dalam variabel `rp_speed`, `rp_accel`, `rp_x`, `rp_y`, `rp_dest_x`, dan `rp_dest_y` pada berkas `aodv_packet.h` seperti ditunjukkan pada Gambar 4.4.

```
116 struct hdr_aodv_reply {
117     u_int8_t rp_type; // Packet Type
118     u_int8_t reserved[2];
119     u_int8_t rp_hop_count; // Hop Count
120     nsaddr_t rp_dst; // Destination IP Address
121     u_int32_t rp_dst_seqno; // Destination Sequence Number
122     nsaddr_t rp_src; // Source IP Address
123     double rp_lifetime; // Lifetime
124
125     double rp_timestamp; // when corresponding REQ sent;
126     // used to compute route discovery latency
127
128     // Modifikasi hello message header
129     double rp_speed, rp_accel, rp_x, rp_y, rp_dest_x,
rp_dest_y;
```

Gambar 4.4. Penambahan atribut pada berkas `aodv_packet.h`

Pada berkas aodv.h perlu ditambahkan variabel lastUpdateTime, lastSpeed, dan lastAccel. Tiga variabel tersebut diperlukan untuk menyimpan informasi kecepatan dan percepatan terakhir dari sebuah node. Informasi kecepatan dan waktu update terakhir digunakan untuk menghitung nilai percepatan berikutnya dari sebuah node. Modifikasi berkas aodv.h dapat dilihat pada Gambar 4.5.

```

nsaddr_t      index;           // IP Address of this node
u_int32_t     seqno;           // Sequence Number
int           bid;             // Broadcast ID

// Modifikasi
double        lastUpdateTime;
double        lastAccel;
double        lastSpeed;

```

Gambar 4.5. Modifikasi berkas aodv.h

Pada AODV-PNT, informasi nilai faktor kecepatan, percepatan, lokasi, dan arah dari semua node tetangga akan disimpan pada kelas AODV_Neighbor dalam berkas aodv_rtable.h. Modifikasi yang dilakukan dapat dilihat pada Gambar 4.6.

```

class AODV_Neighbor {
    friend class AODV;
    friend class aodv_rt_entry;
public:
    AODV_Neighbor(u_int32_t a) { nb_addr = a; }

protected:
    LIST_ENTRY(AODV_Neighbor) nb_link;
    nsaddr_t      nb_addr;
    double        nb_expire;
    // Modifikasi
    double nb_speed, nb_accel, nb_x, nb_y, nb_dest_x,
    nb_dest_y;
};

```

Gambar 4.6. Modifikasi berkas aodv_rtable.h

4.3.3 Modifikasi berkas aodv.cc

Modifikasi yang dilakukan pada berkas aodv.cc terdiri dari modifikasi pada fungsi sendHello dan fungsi recvHello. Pada fungsi sendHello, modifikasi yang dilakukan adalah memasukkan informasi

nilai faktor kecepatan, percepatan, posisi, dan arah dari sebuah node ke dalam pesan Hello. Kode sumber modifikasi pada fungsi sendHello dapat dilihat pada Lampiran 1. Pseudocode modifikasi pada fungsi sendHello ditampilkan pada Gambar 4.7.

Pseudocode Modifikasi fungsi sendHello

```

1: iNode = getMobileNode(nodeID)
2: set Hello message's Speed, Coordinate, and Destination
   as iNode's
3: if now is equal to previous update time
4:     then Hello message's Accel = previous Accel
5: else
6:     Hello message's Accel = (Speed - previous Speed)
       / (now - previous update time)
7:     previous Accel = Hello message's Accel
8:     previous Speed = Hello message's Speed
9: end if

```

Gambar 4.7 Pseudocode modifikasi fungsi sendHello

Pada fungsi recvHello, modifikasi yang dilakukan adalah menyimpan informasi nilai faktor-faktor dari node tetangga di dalam pesan Hello ke dalam kelas neighbor. Modifikasi ini ditunjukkan pada Gambar 4.8.

```

void AODV::recvHello(Packet *p) {
    struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
    AODV_Neighbor *nb;

    nb = nb_lookup(rp->rp_dst);
    if(nb == 0) {
        nb_insert(rp->rp_dst);
        // Modifikasi
        nb = nb_lookup(rp->rp_dst);
        nb->nb_speed = rp->rp_speed;
        nb->nb_accel = rp->rp_accel;
        nb->nb_x      = rp->rp_x;
        nb->nb_y      = rp->rp_y;
        nb->nb_dest_x = rp->rp_dest_x;
        nb->nb_dest_y = rp->rp_dest_y;
    }
    else {
        // Modifikasi
        nb->nb_speed = rp->rp_speed;
        nb->nb_accel = rp->rp_accel;
        nb->nb_x      = rp->rp_x;
    }
}

```

```

nb->nb_y      = rp->rp_y;
nb->nb_dest_x = rp->rp_dest_x;
nb->nb_dest_y = rp->rp_dest_y;

```

Gambar 4.8. Modifikasi pada fungsi recvHello

4.3.4 Perhitungan nilai TWR dan future TWR

Perhitungan nilai TWR dan future TWR dilakukan pada saat sebuah node mengirimkan atau menerima paket request. Perhitungan tersebut diimplementasikan pada fungsi sendRequest dan recvRequest pada berkas aodv.cc. Kode sumber perhitungan TWR dapat dilihat pada Lampiran 2. Pseudocode perhitungan TWR ditampilkan pada Gambar 4.9. Kode sumber perhitungan future TWR ditunjukkan pada Lampiran 3. Pseudocode perhitungan future TWR ditampilkan pada Gambar 4.10.

Pseudocode Perhitungan TWR

```

1: for every neighbor in neighbor list
2:   Distance = Cartesian Distance (destination
                                   coordinate, neighbor coordinate)
3:   Radius = Min ( Cartesian Distance (this node
                                       coordinate, neighbor coordinate ), max
                                       transmission range )
4:   Quality = 1 / ( 1 - ( Radius / ( max transmission
                                   range +1)))
5:   Direction = getDirection ( neighbor delta x,
                              neighbor delta y)
6:   TWR = (Speed * fSpeed) + (Accel * fAccel) +
          (Distance * fDistance) + (Quality * fQuality) +
          (Direction * fDirection)
7: End for

```

Gambar 4.9. Pseudocode perhitungan TWR

Pseudocode Perhitungan Future TWR

```

1: for every neighbor in neighbor list
2:   Future speed = Speed + (Accel * Delta time)
3:   Future neighbor's x coordinate = neighbor_x + (
neighbor_Speed * Delta time) + (0.5 * neighbor_Accel *
Delta time^2)
4:   Future neighbor's y coordinate = neighbor_y + (
neighbor_Speed * Delta time) + (0.5 * neighbor_Accel *
Delta time^2)
5:   Future x coordinate = x + (Speed * Delta time) + (0.5 *
Accel * Delta time^2)
6:   Future y coordinate = y + (Speed * Delta time) + (0.5 *
Accel * Delta time^2)
7:   Future Distance = Cartesian Distance (coordinate,
neighbor future coordinate)

```

```

8: Future Radius = Min ( Cartesian Distance (this node
future coordinate, neighbor future coordinate ), max
transmission range )
9: Future Quality = 1 / ( 1 - ( Future Radius / ( max
transmission range +1)))
10: Future TWR = (Future Speed * fSpeed) + (Accel * fAccel)
+ (Future Distance * fDistance) + (Future Quality *
fQuality) + (Direction * fDirection)
11: End for

```

Gambar 4.10. Pseudocode perhitungan future TWR

Pada penelitian ini, salah satu modifikasi yang dilakukan terhadap AODV-PNT adalah membedakan bobot faktor arah berdasarkan apakah kedua node bergerak saling mendekat atau menjauh. Kode sumber perhitungan faktor arah dapat dilihat pada Lampiran 4.

Modifikasi berikutnya adalah penerapan nilai ambang batas untuk masing-masing faktor dalam penentuan stabilitas node. Untuk memperjelas, penulis tampilkan kembali rumus perhitungan TWR sebagai berikut:

$$TWR = f_s \times |S_n - S_d| + f_a \times |A_n - A_d| + f_d \times |\Theta_n - \Theta_d| + f_q \times Q + f_{dist} \times Dist$$

S_n , A_n , dan Θ_n : Kecepatan, percepatan, dan arah dari next-hop node.

S_d , A_d , dan Θ_d : Kecepatan, percepatan, dan arah dari node tujuan.

f_s : bobot pengali kecepatan.

f_a : bobot pengali percepatan.

f_d : bobot pengali arah.

f_q : bobot pengali kualitas tautan.

Q : kualitas tautan antara node sumber dengan next-hop node.

f_{dist} : bobot pengali jarak.

$Dist$: jarak antara next-hop node dengan node tujuan.

Pada AODV-PNT mekanisme penentuan stabil tidaknya sebuah node didasarkan pada perbandingan antara nilai total ΔTWR (selisih mutlak nilai TWR dengan future TWR) dengan nilai ambang

batas W . Jika nilai ΔTWR kurang dari W , maka node tersebut dinyatakan stabil. Pada penelitian ini, nilai ambang batas diterapkan pada masing-masing faktor, sehingga ada nilai W_{speed} untuk kecepatan, $W_{quality}$ untuk kualitas tautan, dan $W_{distance}$ untuk jarak. Nilai untuk faktor percepatan dan arah diasumsikan tidak berubah ketika perhitungan future TWR, sehingga tidak ada nilai ambang batas untuk kedua faktor tersebut. Jadi, sebuah node akan dianggap stabil jika dan hanya jika:

$$| \text{future } S_n - \text{future } S_d | \leq W_{speed}$$

$$\&\& \text{future } Q \leq W_{quality}$$

$$\&\& \text{future } Dist \leq W_{distance}$$

Implementasi penerapan nilai ambang batas dapat dilihat pada Lampiran 5.

4.4. Implementasi Metrik Analisis

Simulasi pada NS-2 menghasilkan sebuah berkas hasil simulasi yang disebut trace file dengan ekstensi .tr. Trace file dapat dianalisa menggunakan skrip AWK untuk mengukur performa dari protokol yang disimulasikan. Ada tiga skrip AWK yang diimplementasikan pada penelitian ini, masing-masing untuk mengukur *packet delivery ratio*, *average delay*, dan *routing overhead*.

Packet delivery ratio adalah perbandingan dari jumlah paket yang dikirimkan dan diterima. Jumlah paket dihitung berdasarkan baris pada trace file yang mengandung string AGT. String AGT adalah penanda untuk event yang terjadi pada application layer. Skrip AWK untuk perhitungan *packet delivery ratio* dapat dilihat pada Lampiran 6.

Routing overhead adalah jumlah paket routing yang ditransmisikan. Paket routing yang ditransmisikan terdiri dari *route request* (RREQ), *route reply* (RREP), dan *route error* (RERR). Jumlah paket dihitung berdasarkan baris pada trace file yang

mengandung string RREQ, RREP, ataupun RERR. Skrip AWK perhitungan routing overhead dapat dilihat pada Lampiran 7.

Average end-to-end delay merupakan rata-rata waktu yang dibutuhkan oleh semua paket untuk sampai ke tujuan. Paket yang diperhitungkan hanya paket yang mencapai tujuan. Langkah pertama untuk menghitung delay adalah dengan mengelompokkan paket pada trace file berdasarkan ID paket yang terdapat pada kolom keenam. Jenis paket yang diperhitungkan adalah paket CBR. Kemudian, selisih waktu penerimaan dan pengiriman masing-masing paket akan dihitung. Selisih waktu tersebut akan dijumlahkan untuk semua paket, dan dibagi dengan jumlah paket. Skrip AWK untuk perhitungan *Average end-to-end delay* dapat dilihat pada Lampiran 8.

BAB 5

UJI COBA DAN EVALUASI

5.1. Lingkungan Uji Coba

Perangkat keras yang digunakan untuk uji coba mempunyai spesifikasi yang dapat dilihat pada Tabel 5.1. Perangkat lunak untuk lingkungan simulasi adalah Oracle VM VirtualBox dengan spesifikasi yang ditunjukkan pada Tabel 5.2. Simulator yang digunakan adalah NS-2 dengan parameter yang ditunjukkan pada Tabel 5.3.

Tabel 5.1.

Spesifikasi perangkat keras

Processor	Intel Celeron CPU N3350 1.10GHz
Sistem Operasi	Windows 10 Pro 64-bit
RAM	2.00 GB DDR3 L
Penyimpanan	500 GB

Tabel 5.2.

Spesifikasi perangkat lunak

Perangkat Lunak	Oracle VM VirtualBox versi 5.1.18
Sistem Operasi	Linux Ubuntu 64-bit
Processor	1 CPU
Base Memory	1 GB
Video Memory	16 MB

Skenario simulasi dibuat menggunakan SUMO. Skenario simulasi dibedakan menjadi 9 variasi skenario berdasarkan variasi tingkat mobilitas dan densitas. Variasi skenario dapat dilihat pada Tabel 5.4. Setiap variasi terdiri atas 20 skenario (10 skenario grid dan 10 skenario riil). Penelitian ini memakai total 180 skenario. Setiap

skenario memiliki aktifitas dan pergerakan node yang berbeda-beda sesuai dengan yang dihasilkan oleh SUMO secara acak.

Tabel 5.3.

Spesifikasi parameter simulasi NS-2

No.	Parameter	Spesifikasi
1	Simulator	NS-2.35
2	Waktu Simulasi	360 detik
3	Radius Transmisi	250 meter
4	Agen	CBR (Constant Bit Rate)
5	Ukuran Paket	512 Bytes
6	Packet Rate	2 kB/s
7	Packet Interval	1 packet/s
8	Model Propagasi	Two-ray ground
9	Protokol MAC	IEEE 802.11p

Tingkat mobilitas dibagi dalam tiga kategori. Tingkat mobilitas rendah mempunyai kecepatan 2 – 14 m/s, tingkat mobilitas sedang 15 – 22 m/s, dan tingkat mobilitas tinggi 23 – 28 m/s. Tingkat densitas juga dibagi dalam tiga kategori. Tingkat densitas rendah memiliki *node* sebanyak 30, tingkat densitas sedang 50, dan tingkat densitas tinggi 100. Variasi tingkat densitas dan mobilitas dapat dilihat pada Tabel 3.3.

Tabel 5.4.

Variasi skenario berdasarkan tingkat mobilitas dan densitas

Variasi	Mobilitas	Densitas
1	Rendah	Rendah
2	Rendah	Sedang
3	Rendah	Tinggi
4	Sedang	Rendah
5	Sedang	Sedang

6	Sedang	Tinggi
7	Tinggi	Rendah
8	Tinggi	Sedang
9	Tinggi	Tinggi

Ada lima uji coba pada penelitian ini. Daftar uji coba yang dilakukan serta tujuan masing-masing uji coba dapat dilihat pada Tabel 5.5.

Tabel 5.5.

Daftar uji coba

Nomor	Uji Coba	Tujuan
1	Pencarian nilai bobot optimal	Mencari nilai bobot optimal masing-masing faktor untuk setiap variasi skenario
2	Pembedaan bobot faktor arah	Mengetahui apakah pembedaan bobot faktor arah berdasarkan saling menjauh atau mendekatnya node meningkatkan performa metode yang diusulkan
3	Pencarian nilai ambang batas optimal	Mencari nilai ambang batas optimal masing-masing faktor untuk setiap variasi skenario
4	Uji coba skenario grid	Membandingkan performa metode yang diusulkan dengan AODV dan AODV-PNT pada peta berbentuk grid. Peta grid digunakan sebagai tes awal implementasi karena bentuknya yang teratur dan seimbang
5	Uji coba skenario riil	Membandingkan performa metode yang diusulkan dengan AODV dan AODV-PNT pada peta riil. Peta riil digunakan untuk melihat performa metode jika diimplementasikan di dunia nyata

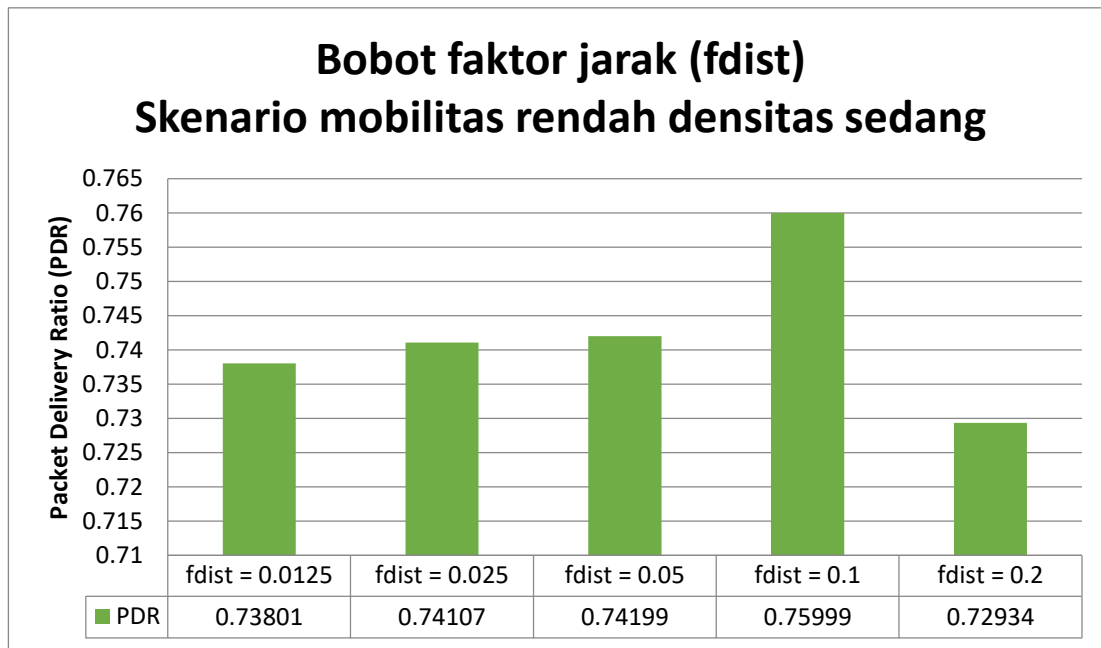
5.2. Uji Coba Pencarian Nilai Bobot Optimal Masing-masing Faktor

Salah satu modifikasi pada penelitian ini adalah penerapan bobot dinamis pada faktor-faktor yang mempengaruhi pemilihan rute (kecepatan, percepatan, arah, jarak, kualitas tautan). Bobot masing-masing faktor tidak selalu sama untuk variasi mobilitas dan densitas yang berbeda. Subbab ini menyajikan detail uji coba pencarian nilai bobot optimal untuk masing-masing faktor untuk setiap variasi skenario.

Untuk uji coba ini, yang harus dilakukan pertama kali adalah menentukan nilai bobot awal untuk masing-masing faktor. Langkah untuk penentuan nilai bobot awal untuk masing-masing faktor adalah sebagai berikut:

1. Catat semua selisih nilai faktor antara sebuah node dengan node tetangganya pada fungsi `sendRequest` dan `recvRequest` pada berkas `aodv.cc`.
2. Jumlahkan semua nilai tersebut untuk semua skenario yang memiliki jenis variasi densitas dan mobilitas yang sama, lalu cari nilai rata-ratanya.
3. Ulangi sampai didapatkan nilai rata-rata untuk masing-masing faktor kecepatan, percepatan, jarak, arah, dan kualitas tautan.
4. Kalikan nilai rata-rata per faktor dengan sebuah bilangan sehingga nilai semua faktor menjadi hampir sama. Contoh untuk variasi skenario mobilitas rendah densitas sedang (variasi 2), nilai rata-rata faktor kecepatan, percepatan, jarak, kualitas tautan, dan arah secara berurutan adalah 7.5, 0.51, 373, 23.45, dan 97. Nilai-nilai tersebut kemudian dikalikan dengan (secara urut) 3, 39, 0.05, 0.85, dan 0.2 sehingga nilai semua faktor menjadi hampir sama yaitu sekitar 20. Nilai bobot awal untuk masing-masing faktor adalah bilangan pengali tadi (3 untuk kecepatan, 39 untuk percepatan, dan seterusnya).
5. Simulasikan dan catat nilai PDR dengan menggunakan nilai bobot awal. Kemudian ubah nilai salah satu faktor dengan cara dikali 2, dibagi 2, dikali 4, dan dibagi 4. Catat PDR untuk masing-masing nilai.

6. Cari nilai PDR tertinggi (titik puncak) dari kelima nilai bobot untuk masing-masing faktor. Nilai bobot optimal adalah nilai bobot yang menghasilkan nilai PDR tertinggi. Contoh grafik hasil percobaan nilai faktor jarak pada skenario variasi 2 ditunjukkan pada Gambar 5.1. Berdasarkan grafik, nilai bobot optimal untuk faktor jarak pada variasi skenario 2 adalah 0.1. Hasil lengkap nilai bobot optimal tiap faktor dapat dilihat pada Tabel 5.5.



Gambar 5.1. Grafik pencarian nilai bobot optimal faktor jarak pada variasi skenario 2

Tabel 5.6.

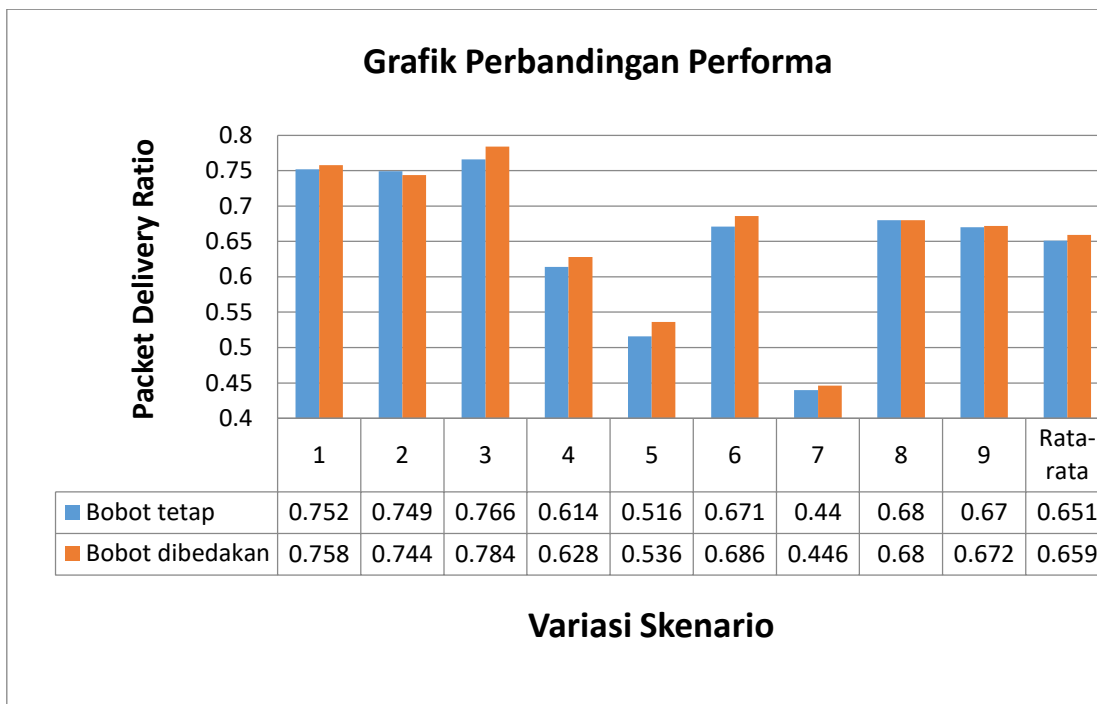
Hasil lengkap nilai bobot optimal tiap faktor

Variasi Skenario	F kecepatan	F percepatan	F jarak	F kualitas tautan	F arah
1	3	20	0.1	2	0.2
2	1.5	156	0.1	1.7	0.2
3	1.5	10	0.2	3.48	0.2
4	5	4.5	0.2	0.15	0.2
5	0.625	4.5	0.2	1.2	0.1
6	3	4.5	0.05	0.7	0.1
7	0.5	56	0.2	0.15	0.4

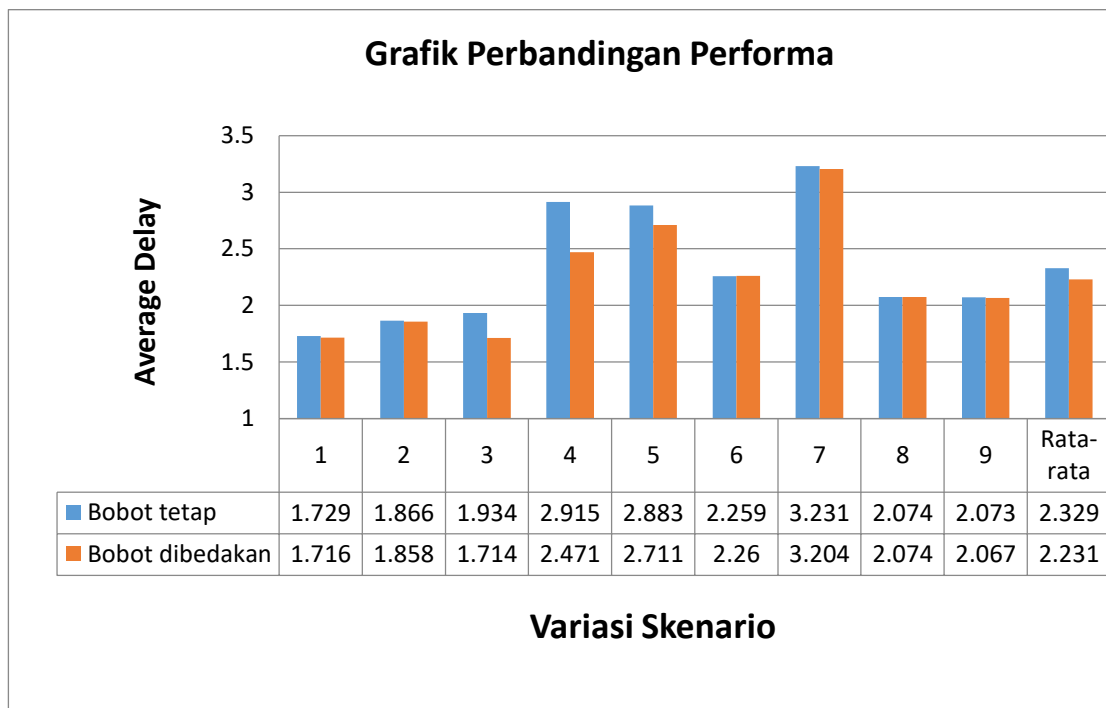
8	4	3.75	0.05	0.15	0.2
9	1.4	8	0.025	2.8	0.8

5.3. Uji Coba Pembedaan Bobot Faktor Arah Berdasarkan Menjauh atau Mendekatnya Node

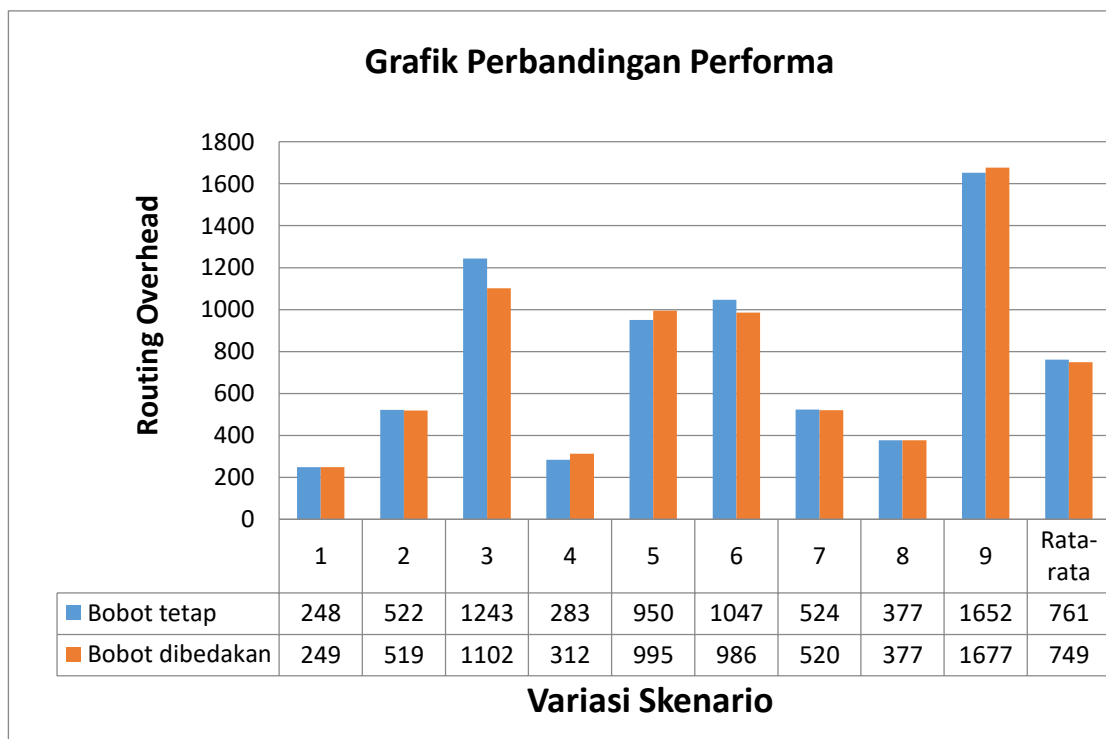
Pada penelitian ini, salah satu modifikasi yang dilakukan terhadap AODV-PNT adalah membedakan bobot pengali faktor arah berdasarkan apakah kedua node bergerak saling mendekat atau menjauh. Apabila kedua node bergerak saling menjauh, maka bobot akan diubah menjadi dua kali lipat daripada bobot apabila kedua node bergerak saling mendekat. Subbab ini menyajikan hasil perbandingan antara performa apabila bobot pengali faktor arah dibedakan berdasarkan mendekat menjauhnya node dengan apabila bobot tidak dibedakan. Hasil perbandingan dapat dilihat pada Gambar 5.2 (PDR), Gambar 5.3 (Delay), dan Gambar 5.4 (Overhead)



Gambar 5.2. Grafik perbandingan PDR bobot tetap dan dibedakan



Gambar 5.3. Grafik perbandingan Delay bobot tetap dan dibedakan



Gambar 5.4. Grafik perbandingan Overhead bobot tetap dan dibedakan

Berdasarkan grafik hasil perbandingan, metode yang diusulkan menghasilkan PDR, Delay, serta Overhead yang lebih baik jika bobot pengali faktor arah untuk node yang saling menjauh dua kali lipat dibandingkan yang saling mendekat. Penyebabnya adalah dua node yang saling mendekat akan berada dalam radius komunikasi dalam waktu yang lebih lama. Selain itu, dua node yang saling mendekat akan menyebabkan selisih faktor jarak antar node semakin mengecil, sehingga nilai TWR akan semakin kecil pula.

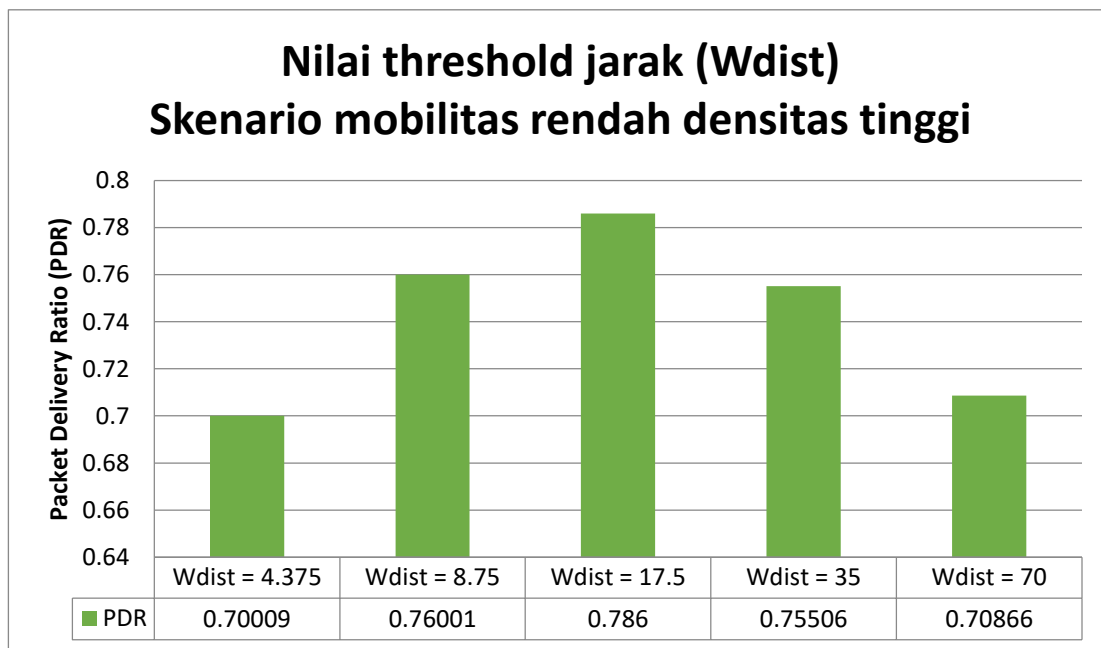
5.4. Uji Coba Pencarian Nilai Ambang Batas Optimal

Modifikasi berikutnya dari penelitian ini adalah penggunaan nilai ambang batas untuk masing-masing Δ nilai ($\text{future TWR} - \text{TWR}$) untuk faktor kecepatan, jarak, dan kualitas tautan. Faktor percepatan dan arah tidak perlu nilai ambang batas karena tidak mempunyai nilai future TWR. Nilai ambang batas masing-masing faktor tidak selalu sama untuk variasi mobilitas dan densitas yang berbeda. Subbab ini menyajikan detail uji coba pencarian nilai ambang batas optimal untuk masing-masing faktor untuk setiap variasi skenario.

Untuk uji coba ini, yang harus dilakukan pertama kali adalah menentukan nilai ambang batas awal untuk masing-masing faktor. Langkah untuk penentuan nilai ambang batas awal untuk masing-masing faktor adalah sebagai berikut:

1. Catat semua selisih absolut nilai future dan nilai saat ini tiap faktor pada fungsi `sendRequest` dan `recvRequest` pada berkas `aodv.cc`.
2. Jumlahkan semua nilai tersebut untuk semua skenario yang memiliki jenis variasi densitas dan mobilitas yang sama, lalu cari nilai rata-ratanya.
3. Ulangi sampai didapatkan nilai rata-rata untuk masing-masing faktor kecepatan, jarak, dan kualitas tautan.
4. Nilai rata-rata tersebut dijadikan untuk nilai awal ambang batas masing-masing faktor. Contoh untuk variasi skenario mobilitas rendah densitas tinggi (variasi 3), nilai selisih absolut rata-rata faktor kecepatan, jarak, dan kualitas tautan secara berurutan adalah 1.5, 17.5, dan 27.

5. Simulasikan dan catat nilai PDR dengan menggunakan nilai ambang batas awal. Kemudian ubah nilai ambang batas salah satu faktor dengan cara dikali 2, dibagi 2, dikali 4, dan dibagi 4. Catat PDR untuk masing-masing nilai.
6. Cari nilai PDR tertinggi (titik puncak) dari kelima nilai ambang batas untuk masing-masing faktor. Nilai ambang batas optimal adalah nilai yang menghasilkan nilai PDR tertinggi. Contoh grafik hasil percobaan nilai ambang batas faktor jarak pada skenario variasi 3 ditunjukkan pada Gambar 5.5. Berdasarkan grafik, nilai ambang batas optimal untuk faktor jarak pada variasi skenario 2 adalah 17.5. Hasil lengkap nilai ambang batas optimal tiap faktor dapat dilihat pada Tabel 5.6.



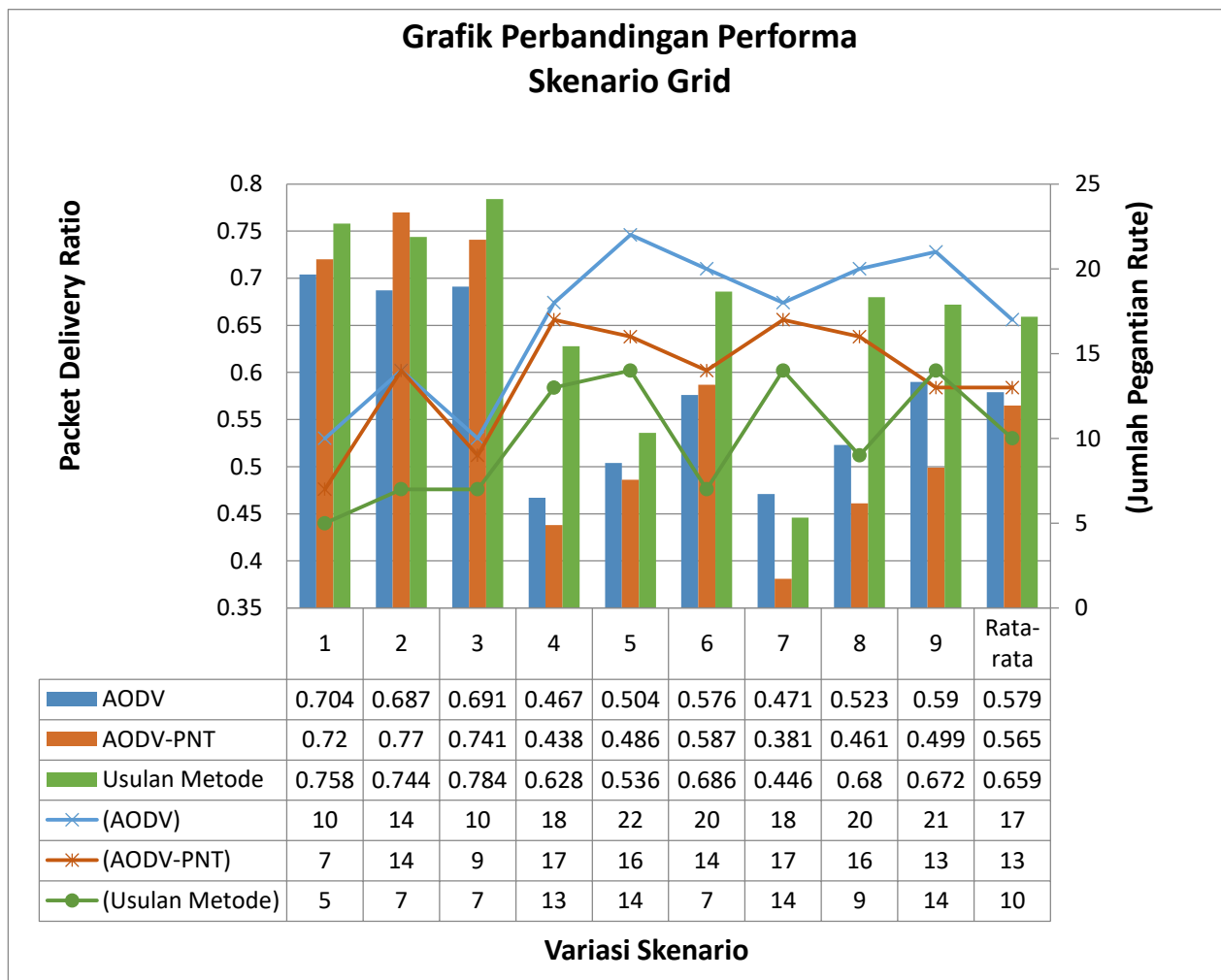
Gambar 5.5. Grafik pencarian nilai ambang batas optimal faktor jarak pada variasi skenario 3

Tabel 5.7.
Hasil lengkap nilai ambang batas optimal tiap faktor

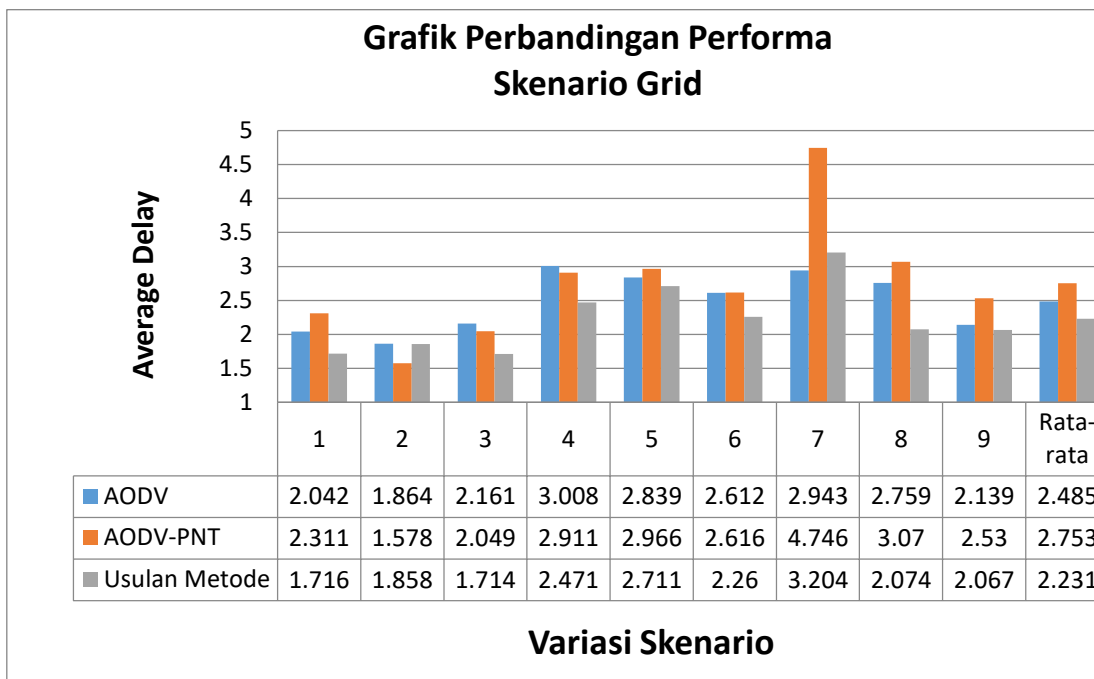
Variasi Skenario	F kecepatan	F jarak	F kualitas tautan
1	0.8	34	27
2	1.6	18	13.5
3	0.375	17.5	6.75
4	0.75	154	36
5	6	164	35
6	1.5	16.5	10.25
7	4	176	22.5
8	1	88	22
9	1	140	23.5

5.5. Uji Coba Skenario Grid

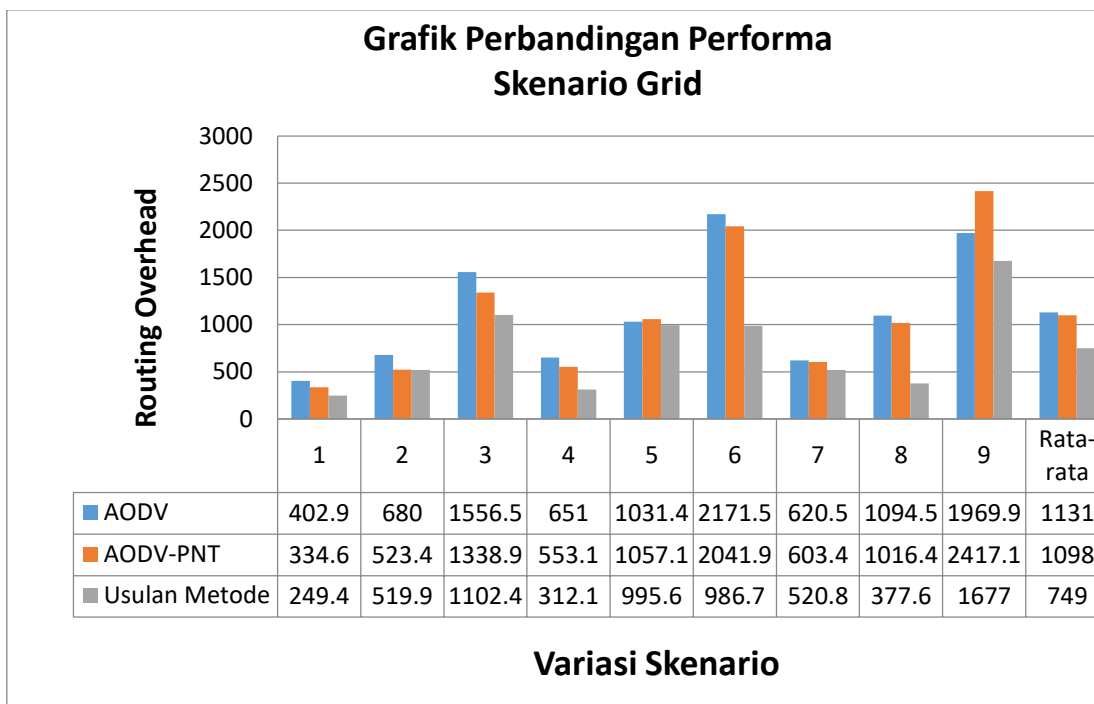
Uji coba skenario grid menggunakan 90 skenario, dengan rincian 10 skenario untuk setiap variasi. Skenario grid dibuat menggunakan SUMO. Setiap skenario dianalisis berdasarkan packet delivery ratio (PDR), average end-to-end delay (Delay) dan routing overhead (Overhead). Grafik untuk PDR, Delay, dan Overhead ditampilkan secara berurutan pada Gambar 5.6, Gambar 5.7, dan Gambar 5.8. Pada setiap grafik, nilai yang ditampilkan adalah rata-rata nilai untuk masing-masing variasi skenario.



Gambar 5.6. Grafik perbandingan PDR skenario grid



Gambar 5.7. Grafik perbandingan Delay skenario grid



Gambar 5.8. Grafik perbandingan Overhead skenario grid

Berdasarkan grafik perbandingan performa, metode yang diusulkan menghasilkan nilai PDR lebih tinggi 9,4% dibandingkan AODV-PNT, dan 8% dibandingkan AODV. Semakin tinggi nilai PDR adalah semakin bagus. Tinggi rendahnya PDR berkaitan dengan stabilitas rute. Semakin stabil rute yang dipilih, semakin tinggi kemungkinan paket mencapai tujuan.

Delay yang dihasilkan oleh metode yang diusulkan lebih rendah 0.52 dibandingkan AODV-PNT dan 0.25 dibandingkan AODV. Semakin rendah nilai Delay adalah semakin bagus. Tinggi rendahnya nilai Delay juga berkaitan dengan stabilitas rute. Semakin banyak pergantian rute, maka waktu tunggu paket untuk mencapai tujuan bertambah, sehingga Delay meningkat.

Overhead yang dihasilkan metode yang diusulkan lebih rendah 349 dari AODV-PNT dan 382 dibandingkan AODV. Semakin rendah nilai overhead semakin bagus. Nilai overhead juga dipengaruhi oleh stabilitas rute. Semakin stabil rute yang dipilih, maka jumlah paket RREP, RREQ, dan RERR yang dikirimkan tentu akan berkurang.

Stabilitas rute yang dipilih oleh metode yang diusulkan dapat dilacak dari trace file yang dihasilkan. Untuk melacak rute, yang harus dilakukan adalah mengambil baris paket yang bertanda REPLY. Contoh dari trace file dapat dilihat pada Gambar 5.9.

```

s 202.013494518 _31_ RTR --- 0 AODV 44 [0 0 0 0] ----- [31:255
30:255 30 26] [0x4 1 [31 126] 10.000000] (REPLY)
f 202.021746562 _26_ RTR --- 0 AODV 44 [13a 1a 1f 800] -----
[31:255 30:255 29 6] [0x4 2 [31 126] 10.000000] (REPLY)
f 206.014976192 _6_ RTR --- 0 AODV 44 [13a 6 1a 800] -----
[26:255 30:255 29 28] [0x4 3 [31 126] 6.000000] (REPLY)
f 206.028110533 _28_ RTR --- 0 AODV 44 [13a 1c 6 800] -----
[26:255 30:255 28 30] [0x4 4 [31 126] 6.000000] (REPLY)
f 206.032271389 _1_ RTR --- 0 AODV 44 [13a 1 1f 800] -----
[31:255 30:255 29 6] [0x4 2 [31 130] 10.000000] (REPLY)
f 206.034269374 _6_ RTR --- 0 AODV 44 [13a 6 1 800] ----- [31:255
30:255 28 28] [0x4 3 [31 130] 10.000000] (REPLY)
f 206.036227374 _28_ RTR --- 0 AODV 44 [13a 1c 6 800] -----
[31:255 30:255 27 30] [0x4 4 [31 130] 10.000000] (REPLY)
r 206.045988507 _30_ RTR --- 0 AODV 44 [13a 1e 1c 800] -----
[31:255 30:255 27 30] [0x4 4 [31 130] 10.000000] (REPLY)

```

```

s 235.018762419 _31_ RTR --- 0 AODV 44 [0 0 0 0] ----- [31:255
30:255 30 19] [0x4 1 [31 138] 10.000000] (REPLY)
f 235.023465400 _19_ RTR --- 0 AODV 44 [13a 13 1f 800] -----
[31:255 30:255 29 5] [0x4 2 [31 138] 10.000000] (REPLY)

```

Gambar 5.9. Contoh trace file

Waktu pengiriman paket didapatkan pada kolom kedua, sedangkan ID dari node-node yang membentuk rute didapatkan dari kolom ketiga. Durasi bertahannya sebuah rute didapatkan dari selisih waktu antara dimulainya sebuah reply baru dari node tujuan dengan waktu reply terakhir diterima oleh node sumber. Pada Tabel 5.7, Tabel 5.8, dan Tabel 5.9 dapat dilihat perbedaan lama bertahannya rute pada AODV, AODV-PNT, dan metode yang diusulkan. Skenario yang digunakan adalah salah satu dari skenario variasi 1.

Tabel 5.8.
Durasi bertahannya rute pada AODV

Rute ke-	Rute (ID node)	Durasi (detik)	Jumlah Paket
1	31 – 1 – 19 – 30	12.98	13
2	31 – 1 – 6 – 28 – 30	20.96	21
3	31 – 19 – 2 – 16 – 30	7.97	8
4	31 – 3 – 2 – 5 – 30	10.96	11
5	31 – 29 – 5 – 30	31.98	32
6	31 – 20 – 6 – 30	6.96	7
7	31 – 14 – 27 – 30	12.97	13
8	31 – 5 – 4 – 30	25.96	26
9	31 – 23 – 7 – 30	10.97	11
10	31 – 13 – 17 – 30	7.95	8

Tabel 5.9.
Durasi bertahannya rute pada AODV-PNT

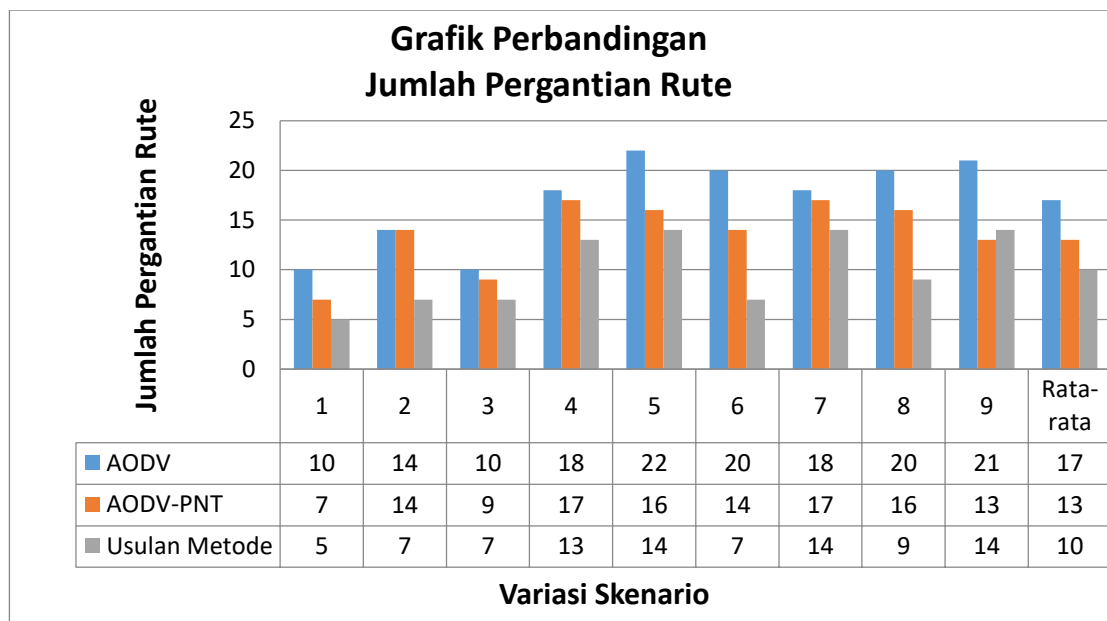
Rute ke-	Rute (ID node)	Durasi (detik)	Jumlah Paket
1	31 – 8 – 6 – 28 – 30	32.96	33
2	31 – 19 – 5 – 30	10.96	11

3	31 – 3 – 5 – 30	17.98	18
4	31 – 29 – 6 – 30	17.98	18
5	31 – 29 – 5 – 4 – 30	52.96	53
6	31 – 4 – 9 – 15 – 30	9.47	10
7	31 – 13 – 23 – 7 – 30	5.44	6

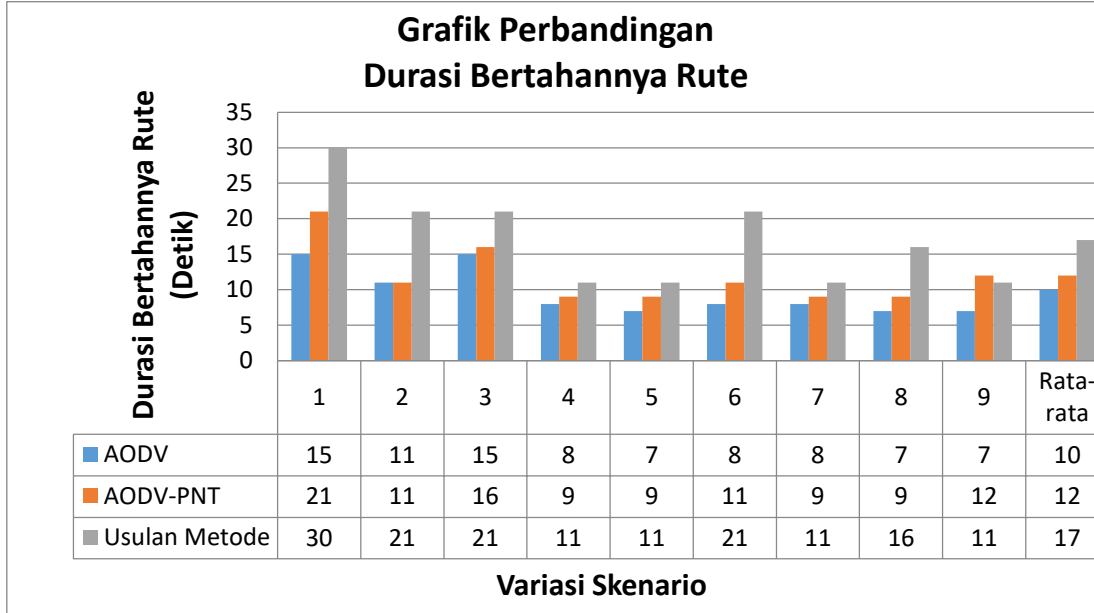
Tabel 5.10.
Durasi bertahannnya rute pada metpode yang diusulkan

Rute ke-	Rute (ID node)	Durasi (detik)	Jumlah Paket
1	31 – 26 – 6 – 28 – 1 – 30	28.97	29
2	31 – 19 – 5 – 30	12.48	13
3	31 – 29 – 5 – 30	40.48	41
4	31 – 21 – 14 – 7 – 27 – 30	20.95	21
5	31 – 14 – 7 – 30	40.97	41

Hasil detail pengujian stabilitas rute ditunjukkan dalam grafik perbandingan jumlah pergantian rute pada Gambar 5.10 dan grafik perbandingan durasi bertahannnya rute pada Gambar 5.11.



Gambar 5.10. Grafik perbandingan jumlah pergantian rute



Gambar 5.11. Grafik perbandingan durasi bertahanannya rute

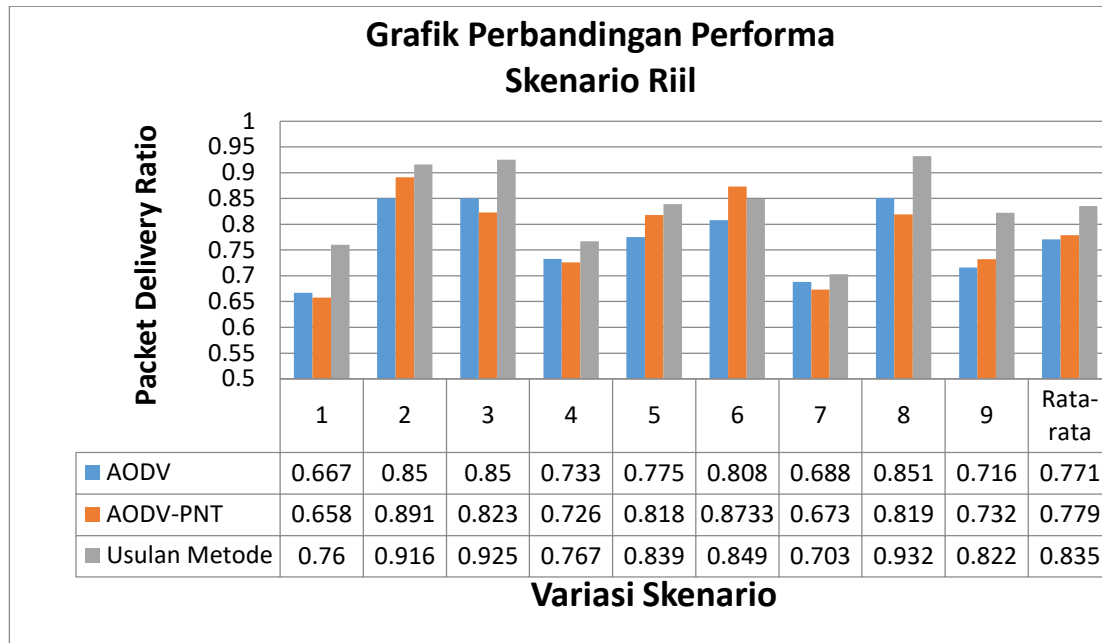
Berdasarkan perbandingan jumlah pergantian rute, AODV adalah yang paling sering berganti rute, sedangkan metode yang diusulkan paling jarang berganti rute. Semakin stabil rute yang dipilih, tentu semakin sedikit terjadi pergantian rute. Berdasarkan perbandingan durasi bertahanannya rute, metode yang diusulkan memiliki durasi paling lama, disusul AODV-PNT, dan terakhir AODV. Dari perbandingan jumlah pergantian rute dan durasi rute, dapat disimpulkan bahwa rute yang dipilih oleh metode yang diusulkan lebih stabil daripada AODV-PNT dan AODV.

5.6. Uji Coba Skenario Riil

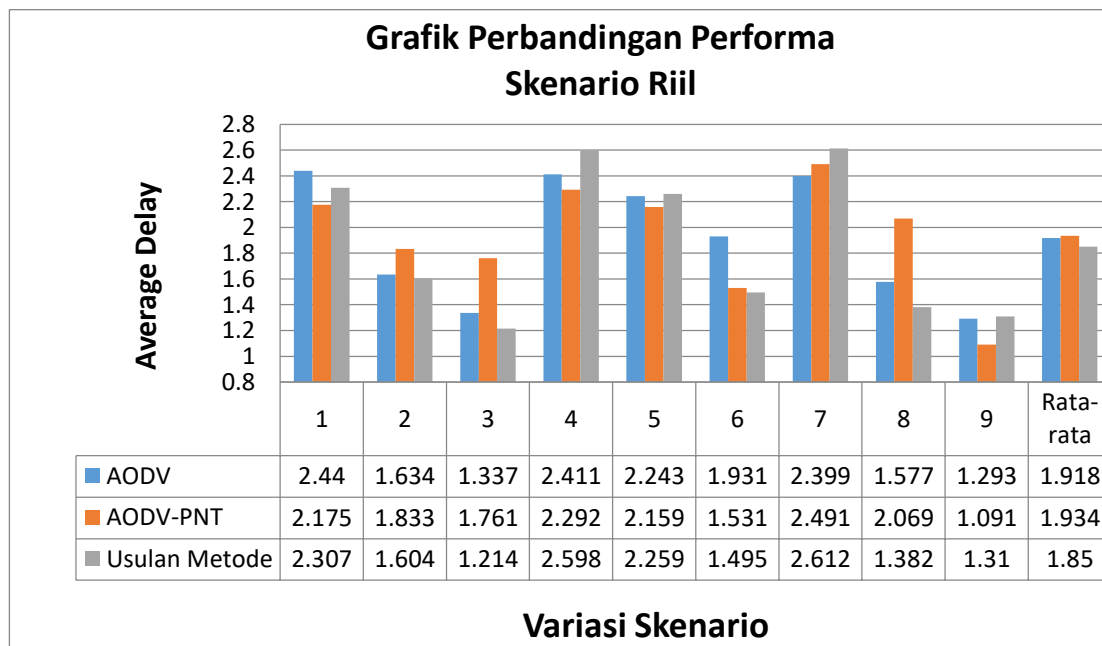
Uji coba skenario riil menggunakan 90 skenario, dengan rincian 10 skenario untuk setiap variasi. Skenario riil menggunakan peta yang diambil dari OpenStreetMap dan diedit menggunakan JOSM editor. Peta yang sudah diedit kemudian dikonversi menjadi berkas .net.xml agar dapat diproses oleh SUMO. Aktivitas dan pergerakan node dihasilkan secara acak oleh SUMO.

Setiap skenario dianalisis berdasarkan packet delivery ratio (PDR), average end-to-end delay (Delay) dan routing overhead (Overhead). Grafik untuk PDR,

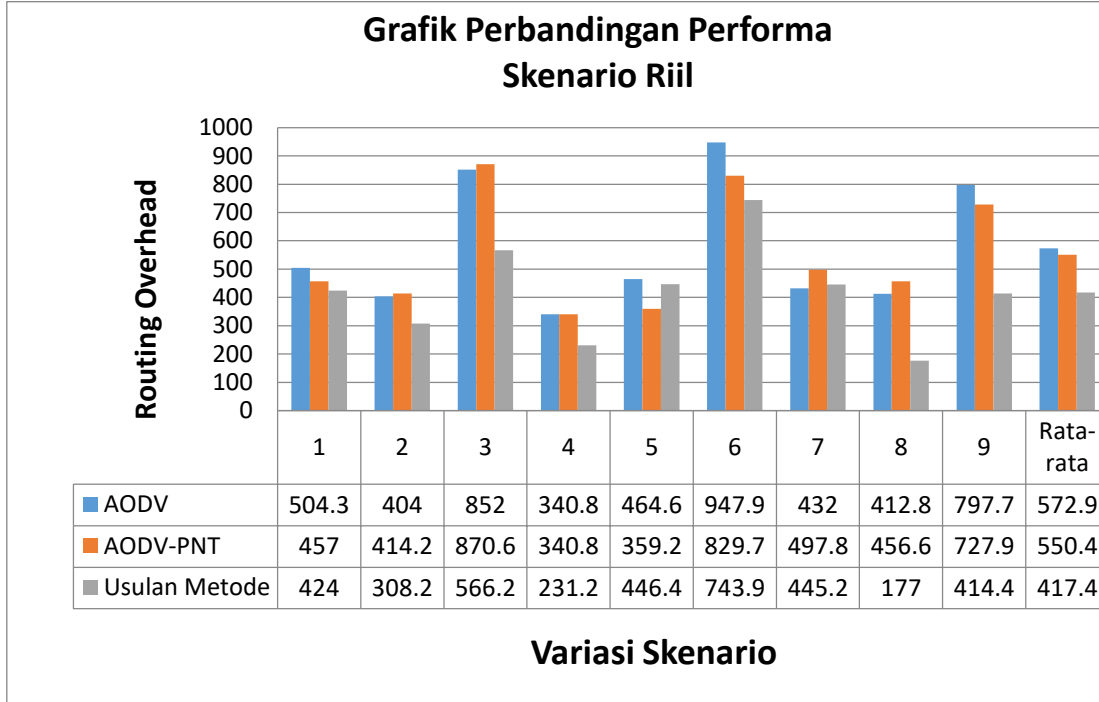
Delay, dan Overhead ditampilkan secara berurutan pada Gambar 5.12, Gambar 5.13 dan Gambar 5.14. Pada setiap grafik, nilai yang ditampilkan adalah rata-rata nilai untuk masing-masing variasi skenario.



Gambar 5.12. Grafik perbandingan PDR skenario riil



Gambar 5.13. Grafik perbandingan Delay skenario riil



Gambar 5.14. Grafik perbandingan Delay skenario riil

Berdasarkan grafik perbandingan performa, metode yang diusulkan menghasilkan nilai PDR lebih tinggi 5.6% dibandingkan AODV-PNT, dan 6.4% dibandingkan AODV. Delay yang dihasilkan oleh metode yang diusulkan lebih rendah 0.08 dibandingkan AODV-PNT dan 0.06 dibandingkan AODV. Overhead yang dihasilkan metode yang diusulkan lebih rendah 133 dari AODV-PNT dan 152 dibandingkan AODV.

Pengujian pada skenario grid dan riil menghasilkan kesimpulan yang sama, yaitu metode yang dihasilkan mempunyai nilai PDR, Delay, dan Overhead yang lebih baik dibandingkan AODV-PNT dan AODV. Hal ini disebabkan oleh beberapa faktor yaitu:

1. Metode yang diusulkan melakukan multicast paket ke sekumpulan relay node, sedangkan AODV melakukan broadcast ke semua next-hop node.

2. Metode yang diusulkan tidak memilih node yang tidak stabil dan memiliki nilai future TWR yang lebih buruk sebagai pilihan next-hop node.
3. Metode yang diusulkan memilih node berdasarkan kemiripan nilai faktor kecepatan, percepatan, arah, kualitas tautan, dan jarak antara next-hop node dengan destinasi sehingga kemungkinan terkirimnya sebuah paket meningkat. AODV-PNT tidak memperhitungkan faktor jarak.
4. Metode yang diusulkan memilih jalur yang lebih stabil dengan cara menyesuaikan bobot faktor kecepatan, percepatan, arah, kualitas tautan, dan jarak antara next-hop node dengan destinasi berdasarkan variasi tingkat mobilitas dan densitas. AODV-PNT menggunakan nilai bobot yang sama untuk semua variasi mobilitas dan densitas.
5. Metode yang diusulkan menerapkan nilai ambang batas untuk perubahan nilai masing-masing faktor, bukan perubahan nilai total semua faktor seperti pada AODV-PNT. Dengan demikian sebuah node yang salah satu nilai faktornya berubah secara drastis akan dianggap sebagai node yang tidak stabil.

[halaman ini sengaja dikosongkan]

BAB 6

KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang dapat diambil berdasarkan hasil pengujian metode yang diusulkan. Bab ini juga berisi saran-saran untuk pengembangan metode yang diusulkan pada masa yang akan datang.

6.1. Kesimpulan

Berdasarkan hasil pengujian dan analisis, kesimpulan yang dapat ditarik adalah sebagai berikut:

1. Metode yang diusulkan memiliki performa yang lebih baik berdasarkan *packet delivery ratio* (PDR), *average end-to-end delay* (Delay), dan *routing overhead* (Overhead) daripada AODV-PNT. PDR yang dihasilkan lebih tinggi 9.4% untuk skenario grid dan 5.6% untuk skenario riil. Delay yang dihasilkan lebih rendah 0.52 dan 0.08 detik. RO yang dihasilkan lebih rendah 349 dan 133 paket. Metode yang diusulkan menghasilkan performa yang lebih baik karena memilih rute yang lebih stabil. Kestabilan rute telah dibuktikan melalui pengujian jumlah pergantian rute dan durasi bertahanya rute. Rata-rata jumlah pergantian rute metode yang diusulkan adalah 10 kali, sedangkan AODV-PNT 13 kali. Rata-rata durasi bertahanya rute metode yang diusulkan adalah 17 detik sedangkan AODV-PNT 12 detik.
2. Nilai bobot optimal untuk masing-masing faktor yang mempengaruhi pemilihan rute berubah-ubah, menyesuaikan dengan variasi densitas dan mobilitas.
3. Perbedaan nilai bobot faktor arah bergantung saling mendekat atau menjauhnya dua buah node mempengaruhi performa metode yang diusulkan. Berdasarkan hasil pengujian, jika bobot node yang saling menjauh dijadikan dua kali lipat daripada yang saling mendekat, performa yang dihasilkan akan

lebih baik, yaitu PDR lebih tinggi 0,8%, Delay lebih rendah 0,098 detik, dan Overhead lebih rendah 12 paket.

4. Nilai ambang batas optimal untuk masing-masing faktor yang mempengaruhi pemilihan rute berubah-ubah, menyesuaikan dengan variasi densitas dan mobilitas. Metode yang diusulkan menerapkan nilai ambang batas untuk perubahan nilai masing-masing faktor, bukan perubahan nilai total semua faktor seperti pada AODV-PNT. Dengan demikian sebuah node yang salah satu nilai faktornya berubah secara drastis akan dianggap sebagai node yang tidak stabil.

6.2. Saran

Saran yang dapat diberikan adalah sebagai berikut. Pemilihan eligible node baik pada AODV-PNT maupun metode yang diusulkan masih memiliki kelemahan. Node-node tetangga dari node sumber akan dieliminasi apabila dianggap tidak stabil. Padahal, bisa jadi node yang tereliminasi tersebut nantinya dapat membentuk jalur yang secara keseluruhan lebih stabil daripada jalur yang dibentuk oleh node yang tidak dieliminasi. Atau bisa jadi sebuah node yang masuk di dalam eligible node ternyata tidak memiliki node tetangga yang bisa mencapai tujuan. Diperlukan sebuah metode untuk mengatasi kelemahan ini.

DAFTAR PUSTAKA

- [1] Hartenstein, Hannes, and L. P. Laberteaux. "A tutorial survey on vehicular ad hoc networks." *IEEE Communications magazine* 46, no. 6 (2008).
- [2] Batool, Fatima, and Shoad A. Khan. "Traffic estimation and real time prediction using adhoc networks." In *Proceedings of the IEEE Symposium on Emerging Technologies*, 2005, pp. 264-269. IEEE, 2005.
- [3] Yu, Xi, Huaqun Guo, and Wai-Choong Wong. "A reliable routing protocol for VANET communications." In *2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1748-1753. IEEE, 2011.
- [4] Artimy, Maen M., William Robertson, and William J. Phillips. "Assignment of dynamic transmission range based on estimation of vehicle density." In *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pp. 40-48. ACM, 2005.
- [5] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. *Ad hoc on-demand distance vector (AODV) routing*. No. RFC 3561. 2003.
- [6] Sun, Xi, and Xia-Miao Li. "Study of the Feasibility of VANET and its Routing Protocols." In *4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008. WiCOM'08. pp. 1-4. IEEE, 2008.
- [7] Rak, Jacek. "Providing differentiated levels of service availability in VANET communications." *IEEE Communications Letters* 17, no. 7 (2013): 1380-1383.
- [8] Shen, Xiaowei, Yi Wu, Zhexin Xu, and Xiao Lind. "AODV-PNT: An improved version of AODV routing protocol with predicting node trend in VANET." In *2014 IEEE 7th International Conference on Advanced Infocomm Technology (ICAIT)*, pp. 91-97. IEEE, 2014.
- [9] Misra, Sudip, Isaac Zhang, and Subhas Chandra Misra, eds. *Guide to wireless ad hoc networks*. Springer Science & Business Media, 2009.
- [10] Johnson, David B., David A. Maltz, and Josh Broch. "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." *Ad hoc networking* 5 (2001): 139-172.

- [11] Namboodiri, Vinod, Manish Agarwal, and Lixin Gao. "A study on the feasibility of mobile gateways for vehicular ad-hoc networks." In Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, pp. 66-75. ACM, 2004.
- [12] Liu, Genping, Bu-Sung Lee, Boon-Chong Seet, Chuan-Heng Foh, Kai-Juan Wong, and Keok-Kee Lee. "A routing strategy for metropolis vehicular communications." In International Conference on Information Networking, pp. 134-143. Springer Berlin Heidelberg, 2004.
- [13] Fussler, Holger, Martin Mauve, Hannes Hartenstein, Michael Kasemann, and Dieter Vollmer. "Location-based routing for vehicular ad-hoc networks." ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), vol. 7, no. 1, pp. 47-49, January 2003.
- [14] Santos, Raúl Aquino, Arthur Edwards, R. M. Edwards, and N. Luke Seed. "Performance evaluation of routing protocols in vehicular ad-hoc networks." International Journal of Ad Hoc and Ubiquitous Computing 1, no. 1-2 (2005): 80-91.
- [15] Wang, S. Y., C. C. Lin, Y. W. Hwang, K. C. Tao, and C. L. Chou. "A practical routing protocol for vehicle-formed mobile ad hoc networks on the roads." In Proceedings of the 8th IEEE international conference on intelligent transportation systems, pp. 161-165. 2005.
- [16] Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." Journal of Computer communications 30, no. 14 (2007): 2826-2841.
- [17] Basu, Prithwish, Naved Khan, and Thomas DC Little. "A mobility based metric for clustering in mobile ad hoc networks." In International Conference on Distributed Computing Systems Workshop, pp. 413-418. IEEE, 2001.
- [18] Wu, Jie, and Hailan Li. "A dominating-set-based routing scheme in ad hoc wireless networks." Journal of Telecommunication Systems 18, no. 1 (2001): 13-36.
- [19] Maihofer, Christian. "A survey of geocast routing protocols." IEEE Communications Surveys & Tutorials 6, no. 2 (2004).

- [20] Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Laura Bieker. "Recent development and applications of SUMO-Simulation of Urban MObility." *International Journal On Advances in Systems and Measurements* 5, no. 3&4 (2012).
- [21] Issariyakul, Teerawat, and Ekram Hossain. *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.
- [22] Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 111 Tahun 2015 Tentang Tata Cara Penetapan Batas Kecepatan. http://ppid.dephub.go.id/files/datahubdat/PM_111_Tahun_2015.pdf. Diakses pada 27-02-2017.

[halaman ini sengaja dikosongkan]

LAMPIRAN

1 Kode Sumber Modifikasi Fungsi sendHello

```
void AODV::sendHello() {
    Packet *p = Packet::alloc();
    struct hdr_cmh *ch = HDR_CMH(p);
    struct hdr_ip *ih = HDR_IP(p);
    struct hdr_aodv_reply *rh = HDR_AODV_REPLY(p);
    // AODV-PNT
    MobileNode *iNode;
    iNode = (MobileNode *) (Node::get_node_by_address(index));
    double iSpeed = ((MobileNode *) iNode)->speed();
    double now = ((MobileNode *) iNode)->getUpdateTime();
    double posX = iNode->X();
    double posY = iNode->Y();
    rh->rp_speed = iSpeed;
    rh->rp_x = posX;
    rh->rp_y = posY;
    rh->rp_dest_x = iNode->destX();
    rh->rp_dest_y = iNode->destY();
    // perhitungan percepatan node
    if (now - lastUpdateTime == 0) {
        rh->rp_accel = lastAccel;
    }
    else {
        rh->rp_accel = (iSpeed - lastSpeed) / (now -
lastUpdateTime); // a = v/t
        lastAccel = rh->rp_accel;
        lastSpeed = iSpeed;
    }
}
```

2 Kode Sumber Perhitungan TWR

```
// Perhitungan TWR
// perhitungan jarak next-hop dan destinasi
double nb_distance;
nb_distance = sqrt(pow((nb->nb_x - xDst), 2) + pow((nb-
>nb_y - yDst), 2));
// radius antara node dengan tetangga
double radius = std::min(
    sqrt(pow((nb->nb_x - posX), 2) + pow((nb->nb_y - posY),
2)), (double) maxTxRange);
// perhitungan faktor kualitas tautan
double quality = 1.0 / (1.0 - (radius / ((double)
maxTxRange + 1.0)));
```

```

double modSpeed      = fSpeed * nb->nb_speed;
double modAccel      = fAccel * nb->nb_accel;
double modDistance   = fDistance * nb_distance;
double modQuality     = fQuality * quality;
double modDirection  = fDirection *
abs(dest_sudut_vertikal-sudut_vertikal);
double TWR = modSpeed + modAccel + modDistance +
modQuality + modDirection;

```

3 Kode Sumber Perhitungan Future TWR

```

// Perhitungan future TWR
// Future speed  $v' = v + a \times t$ 
double nb_speedFuture = nb->nb_speed + (nb->nb_accel *
timeModifier);

// Future posisi
// Formula:  $x' = x + v_0 t + 0.5at^2$ 
// Future posisi node tetangga
double nb_xFuture = nb->nb_x +
(nb->nb_speed * timeModifier)
+ (0.5 * nb->nb_accel * timeModifier *
timeModifier);
double nb_yFuture = nb->nb_y +
(nb->nb_speed * timeModifier)
+ (0.5 * nb->nb_accel * timeModifier *
timeModifier);

// Future posisi node ini
double ixFuture = posX +
(iSpeed * timeModifier)
+ (0.5 * iAccel * timeModifier *
timeModifier);
double iyFuture = posY +
(iSpeed * timeModifier)
+ (0.5 * iAccel * timeModifier *
timeModifier);
double futureDistance;
futureDistance = sqrt(pow((nb_xFuture - xDst), 2) +
pow((nb_yFuture - yDst), 2));

// Future radius
double futureRadius = std::min(
sqrt(pow((nb_xFuture - ixFuture), 2) + pow((nb_yFuture
- iyFuture), 2)), (double) maxTxRange);
double futureQuality = 1.0 / (1.0 - (futureRadius /
((double) maxTxRange + 1.0)));

```

```

// Perhitungan future TWR
modSpeed      = fSpeed * nb_speedFuture;
modAccel      = fAccel * nb->nb_accel;
modDistance   = fDistance * futureDistance;
modQuality    = fQuality * futureQuality;
double futureTWR = modSpeed + modAccel + modDistance +
modQuality + modDirection;

```

4 Kode Sumber Perhitungan Faktor Arah

```

int mendekat = 1;
// selisih nilai x dan y kedua node
delta_x = nb->nb_dest_x - nb->nb_x;
delta_y = nb->nb_dest_y - nb->nb_y;
// menghitung sudut pergerakan berdasarkan garis horizontal
if(delta_x!=0)
    tgtheta = delta_y / delta_x; // tgtheta adalah gradient
    dari pergerakan node
else tgtheta = 0;
theta = atan(tgtheta) * 180.0 / 3.14159265;
// menghitung sudut pergerakan berdasarkan garis vertikal
if(delta_x>0&&delta_y>0)sudut_vertikal=90.0-theta;
else if(delta_x>0&&delta_y<0)sudut_vertikal=90.0-theta;
else if(delta_x<0&&delta_y<0)sudut_vertikal=270.0-theta;
else if(delta_x<0&&delta_y>0)sudut_vertikal=270.0-theta;
else if(delta_x==0&&delta_y>0)sudut_vertikal=0.0;
else if(delta_x==0&&delta_y<0)sudut_vertikal=180.0;
else if(delta_y==0&&delta_x>0)sudut_vertikal=90.0;
else if(delta_y==0&&delta_x<0)sudut_vertikal=270.0;
// perhitungan konstanta untuk persamaan garis
c = nb->nb_dest_y - (tgtheta * nb->nb_dest_x);
// pencarian titik potong berdasarkan konstanta dan
gradient pergerakan kedua node
if(delta_x!=0)
{
    x_ptg = (dest_c - c)/(tgtheta - dest_tgtheta);
    y_ptg = (tgtheta*x_ptg) + c;
}
else
{
    x_ptg = nb->nb_dest_x;
    y_ptg = (dest_tgtheta*x_ptg) + dest_c;
}
// penentuan saling menjauh atau saling mendekatnya dua
buah node
if(sudut_vertikal==0||sudut_vertikal==360){ if(y_ptg<=nb-
>nb_y) mendekat=0;
}
else if(sudut_vertikal==90){if(x_ptg<=nb->nb_x)mendekat=0;
}

```

```

    else if(sudut_vertikal==180){if(y_ptg>nb-
>nb_y)mendekat=0;
    }
    else if(sudut_vertikal==270){if(x_ptg>nb-
>nb_x)mendekat=0;
    }
    else if(sudut_vertikal>0&&sudut_vertikal<90){if(x_ptg<=nb-
>nb_x||y_ptg<=nb->nb_y)mendekat=0;
    }
    else
    if(sudut_vertikal>90&&sudut_vertikal<180){if(x_ptg<=nb-
>nb_x||y_ptg>=nb->nb_y)mendekat=0;
    }
    else
    if(sudut_vertikal>180&&sudut_vertikal<270){if(x_ptg>=nb-
>nb_x||y_ptg>=nb->nb_y)mendekat=0;
    }
    else
    if(sudut_vertikal>270&&sudut_vertikal<360){if(x_ptg>=nb-
>nb_x||y_ptg<=nb->nb_y)mendekat=0;
    }
    if(x_ptg<=dest_x_dst||y_ptg>=dest_y_dst)mendekat=0;

```

5 Kode Sumber Penerapan Nilai Ambang Batas

```

container.isStable = false;
    if((abs(nb_speedFuture-nb->nb_speed)<=Wspeed)
&&(abs(futureDistance-nb_distance)<=Wdistance)
&&(abs(futureQuality-quality)<=Wquality)){
container.isStable = true; }

```

6 Skrip AWK Packet Delivery Ratio

```

BEGIN { sendLine = 0; recvLine = 0; }
$0 ~/^s.* AGT/ { sendLine ++ ;}
$0 ~/^r.* AGT/ { recvLine ++ ;}
END {printf "Sent:%d Recv:%d Ratio:%.4f\n", sendLine,
recvLine, (recvLine/sendLine);}

```

7 Skrip AWK Routing Overhead

```

BEGIN { sendLine = 0; recvLine = 0; errLine = 0; }
$0 ~/^s.* \ (REQUEST\)/ { sendLine ++; }
$0 ~/^s.* \ (REPLY\)/ { recvLine ++; }

```

```
$0 ~/^s.* \ (ERROR\)/      { errLine ++;      }
END { printf "Overhead: %d\n", (sendLine + recvLine +
errLine); }
```

8 Skrip AWK Average Delay

```
BEGIN { highest_packet_id = 0; }
{ # Using old trace file format
  action = $1; time = $2; layer = $4; packet_id = $6;
  # Check for the latest packet id
  if ( packet_id > highest_packet_id ) { highest_packet_id =
packet_id; }
  # If start_time of packet_id is empty, then get read time
  if ( start_time[packet_id] == 0 ) { start_time[packet_id]
= time;}
  if ( $7 == "cbr" ) { # If packet is not dropped
    if ( action != "D" ) { # If packet is received
      if ( action == "r" ) { # Get received time
        end_time[packet_id] = time;}}
  # If packet is dropped
  else { # There is no "end time"
    end_time[packet_id] = -1;}}}}
END {
  sigma_duration = 0; count = 0;
  for ( packet_id = 0; packet_id <= highest_packet_id;
packet_id++ )
    { type = packet_type[packet_id]; start =
start_time[packet_id]; end = end_time[packet_id];
    packet_duration = end - start;
    if ( start < end ) { sigma_duration += packet_duration;
count++;}
    }
  if ( count == 0 ) { printf("no_packet_counted\n");}
  else { printf("Average Delay: %.4f\n", sigma_duration /
count);}
}
```

[halaman ini sengaja dikosongkan]

BIODATA PENULIS



Kukuh Indrayana, biasa dipanggil Kukuh, lahir di kota Surabaya, 15 September 1990. Penulis adalah anak ke-2 dari 2 bersaudara, dibesarkan di kota Surabaya, Jawa Timur. Penulis menempuh pendidikan SDN Semolowaru IV (1996 - 2002), SMPN 19 (2002 - 2005), SMAN 16 (2005 - 2008). Pada tahun 2008, penulis mulai mengikuti studi di S1 jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya. Tahun 2013 penulis menyelesaikan studi S1. Tahun 2014 penulis menempuh strata dua di Jurusan Teknik Informatika, Penulis mengambil bidang minat Net Centric Computing (NCC). Penulis dapat dihubungi melalui alamat e-mail kuhind@gmail.com

[halaman ini sengaja dikosongkan]